Arduino hands-on lab

Stefano Sanna
http://www.gerdavax.it - gerdavax@tiscali.it

SARDEGNA DISTRICT

LAB Open MediaCenter

CRS4

LAB Open Media Center
SARDEGNA DISTRICT

Panoramica

- Mobile, embedded, pervasive: cenni di tecnologia e di interaction design

- Arduino e gli altri...

- Introduzione ad Arduino

- Programmazione

- Hands-on projects

LAB Open Media Center
SARDEGNA DISTRICT

Speaker

- Present:
  - Senior Engineer & Java ME Tech Lead @ beeweeb technologies
  - Developer, technical writer & Java community supporter
- Past:
  - Author of "Java Micro Edition", Hoepli Informatica 2007
  - Software Engineer @ CRS4 [1999-2006]
- Other:
  - Tech & food blogger @ http://www.gerdavax.it
  - Co-developer of first italian iPhone e-book

LAB Open Media Center

SARDEGNA DISTRICT

LAB Open Media Center

SARDEGNA DISTRICT

Embedded

LAB Open MediaCenter

SARDEGNA DISTRICT

Embedded

Mobile

LAB Open Media Center

SARDEGNA DISTRICT

Embedded

Mobile

Pervasive

LAB Open Media Center

SARDEGNA DISTRICT

Embedded

Mobile

Pervasive

Sensor Networks

LAB Open MediaCenter

SARDEGNA DISTRICT

Embedded

Interaction Design: dalla tecnologia all'idea di interazione con gli oggetti del quotidiano

Pe

e

Sensor Networks

LAB Open Media Center
SARDEGNA DISTRICT

I/O e logica

- **Input**: dati provenienti da un insieme di sensori, che trasformano una grandezza fisica di interesse in una tensione (sensore analogico) o in una sequenza di bit (sensore digitale)
- **Elaborazione**: algoritmi che valutano i dati in ingresso e producono dei risultati in uscita
- **Output**: sono cambiamenti nell'ambiente circostante il sistema, eseguiti da un insieme di attuatori, ciascuno in grado di trasformare il risultato dell'elaborazione in azioni fisiche

LAB Open Media Center
SARDEGNA DISTRICT

Sistemi embedded

- Arduino:
  - http://www.arduino.cc
- Phidgets:
  - http://www.phidgets.com
- Basic STAMP:
  - http://www.parallax.com
- MAKE Controller:
  - http://www.makezine.com/controller/
- Sun SPOT:
  - http://www.sunspotworld.com

Phidgets

SunSPOT

LAB Open Media Center
SARDEGNA DISTRICT

Sistemi embedded

- LEGO Mindstorms NXT:
  - http://mindstorms.lego.com
- BUGS:
  - http://www.buglabs.net
- Sentilla Perk
  - http://www.sentilla.com

Mindstorms NXT

Sentilla Perk

Requisiti principali

- Affinché possano essere utilizzati per la realizzazione di sistemi interattivi, i moduli devono possedere queste caratteristiche:

  - programmabilità
  - numero adeguato di porte di input e output (I/O)
  - buona dotazione di interfacce di comunicazione di alto livello
  - facile interfacciabilità con altri sistemi
  - piccole dimensioni e ridotto consumo energetico

  - **community**

LAB Open Media Center
SARDEGNA DISTRICT

Perché Arduino

- Estremamente economico

- Facile da programmare e da interfacciare con hardware altrettanto economico

- Flessibile: può essere adattato a molteplici contesti di utilizzo

- Largamente supportato da una numerosa community di designer, tecnici, appassionati

LAB Open Media Center
SARDEGNA DISTRICT

Perché NON Arduino

- Non costituisce una piattaforma integrata che comprenda un sistema di comunicazione, un set di sensori base, una buona dotazione di memoria permanente e una alimentazione indipendente

- Il microcontrollore e il linguaggio non permettono lo sviluppo di applicazioni complesse (se paragonate a sistemi quali SunSPOT o Sentilla)

Introduzione ad Arduino

Caratteristiche hardware e software

LAB Open Media Center

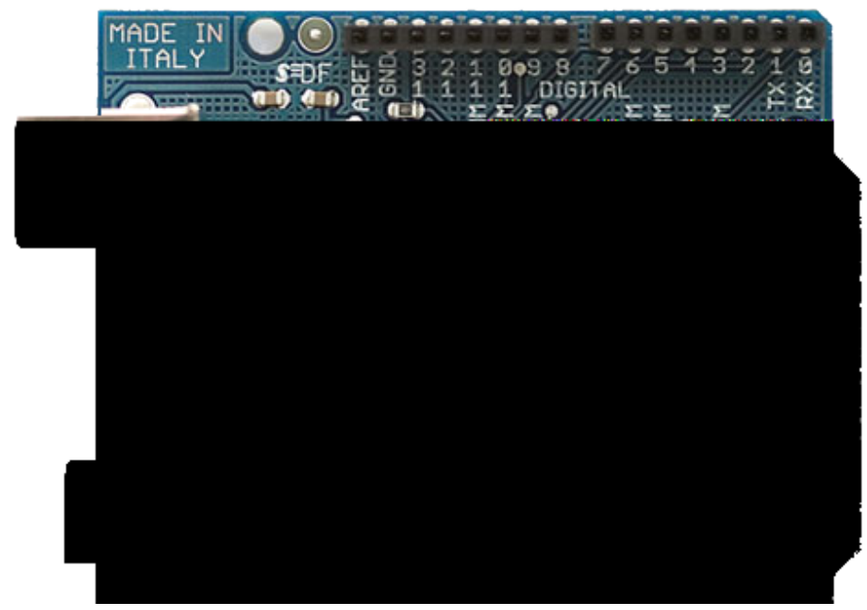
SARDEGNA DISTRICT

CRS4

LAB Open Media Center
SARDEGNA DISTRICT

Arduino

- È un sistema economico, "open", flessibile, semplice da utilizzare, supportato da una vasta comunità di sviluppatori
- Il linguaggio di programmazione deriva dal C
- Arduino è orgogliosamente MADE IN ITALY!
- È sufficiente effettuare una ricerca su Google, Youtube, Flickr per scoprire un ricchissimo patrimonio di idee, progetti, esperimenti, prodotti basati su Arduino



LAB Open Media Center
SARDEGNA DISTRICT

What is Arduino?

- Dal sito ufficiale **http://www.arduino.cc**:

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators.

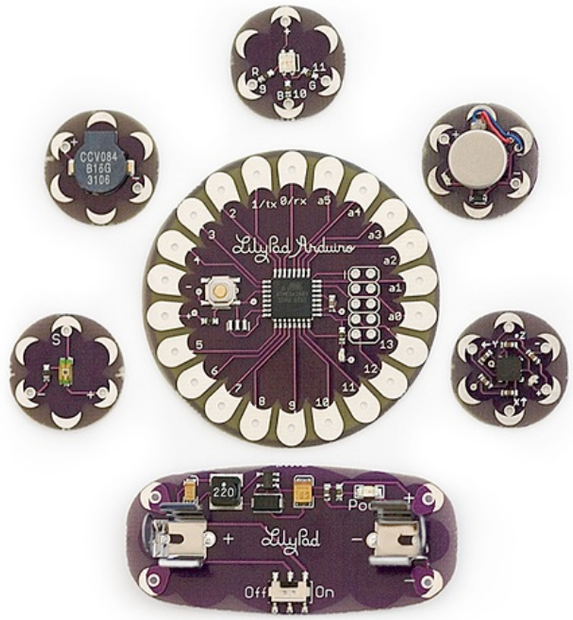
LAB Open Media Center
SARDEGNA DISTRICT

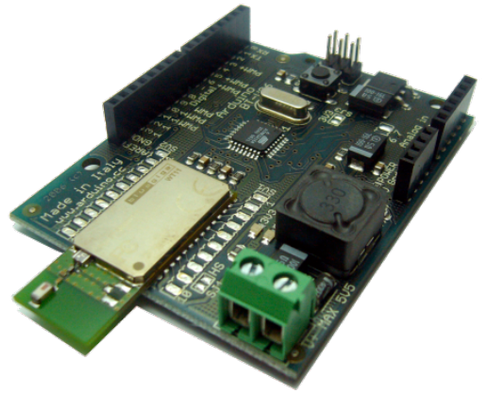
Genesi

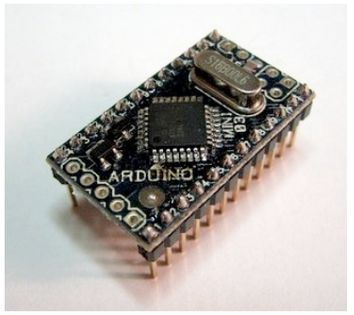
- Arduino è ideato da Massimo Banzi e sviluppato insieme a David Cuartielles, Tom Igoe, Gianluca Martino e David A. Mellis;
- Arduino (linguaggio e piattaforma hardware) deriva dal progetto Wiring, mentre l'ambiente di sviluppo deriva da Processing, un sistema di programmazione per designer
- In breve tempo (tre anni!) Arduino è diventato una degli strumenti di sperimentazione di interaction design più utilizzati al mondo!

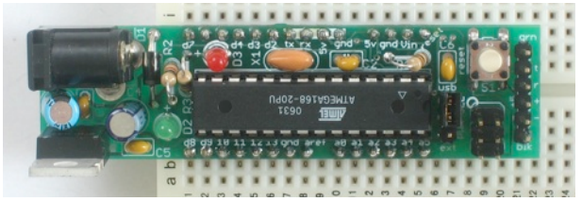
Varianti di Arduino

- Prime versioni
  - Arduino, Arduino USB, Arduino NG
- Versioni attuali:
  - **Arduino Diecimila**
  - Arduino Mini
  - Arduino Nano
  - **Arduino Bluetooth**
  - Lilypad
  - Boarduino, Freeduino





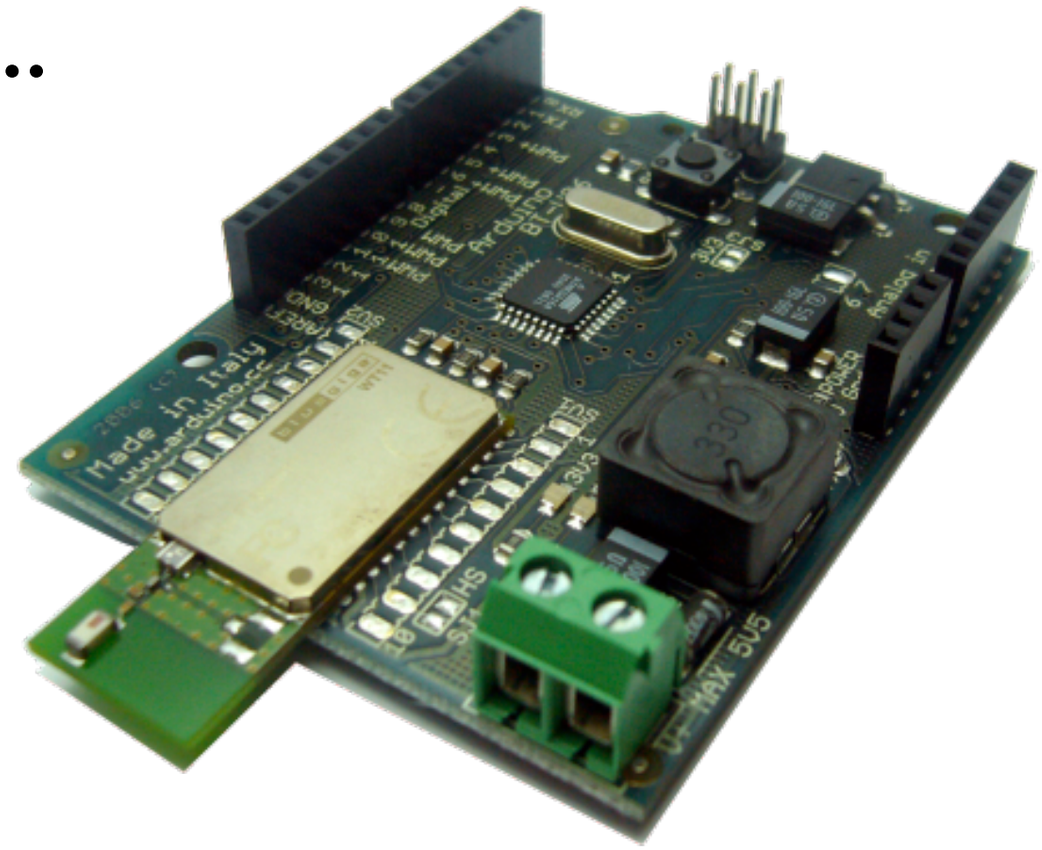




LAB Open Media Center
SARDEGNA DISTRICT

Arduino Bluetooth

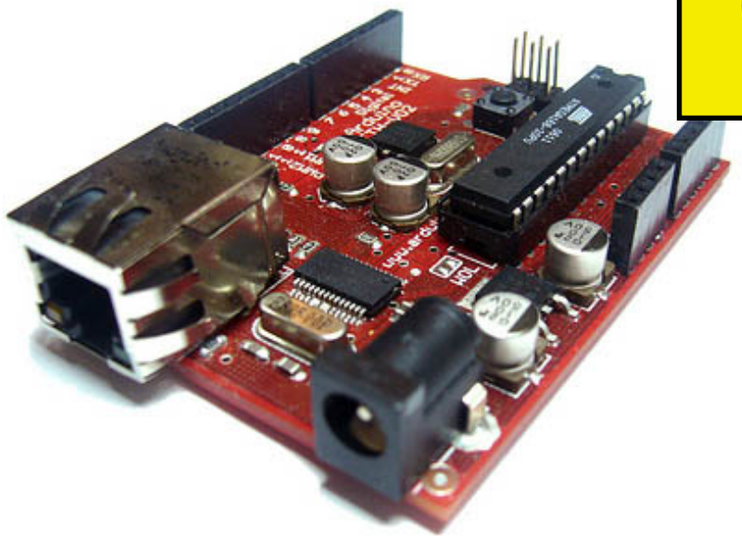
- È equipaggiato di un modulo Bluetooth che sostituisce l'interfaccia USB
- La connessione wireless può essere utilizzata per la programmazione del modulo e per comunicare con periferiche Bluetooth **master** quali computer, cellulari, palmari...



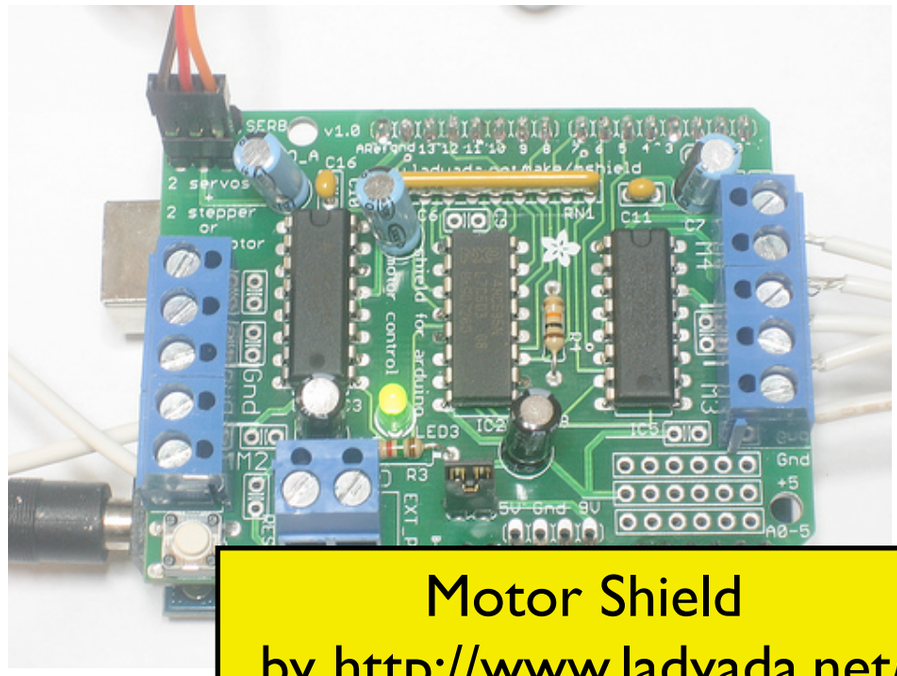
LAB Open MediaCenter

SARDEGNA DISTRICT

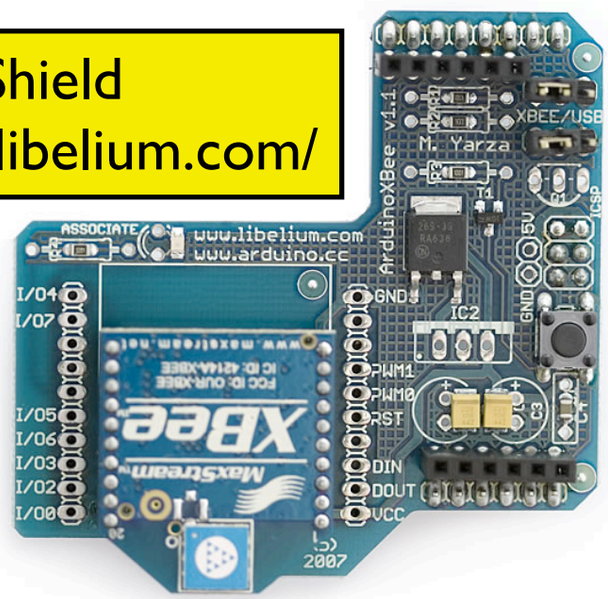
Estensioni



Arduino Ethernet
(early prototype)
by tinker.it

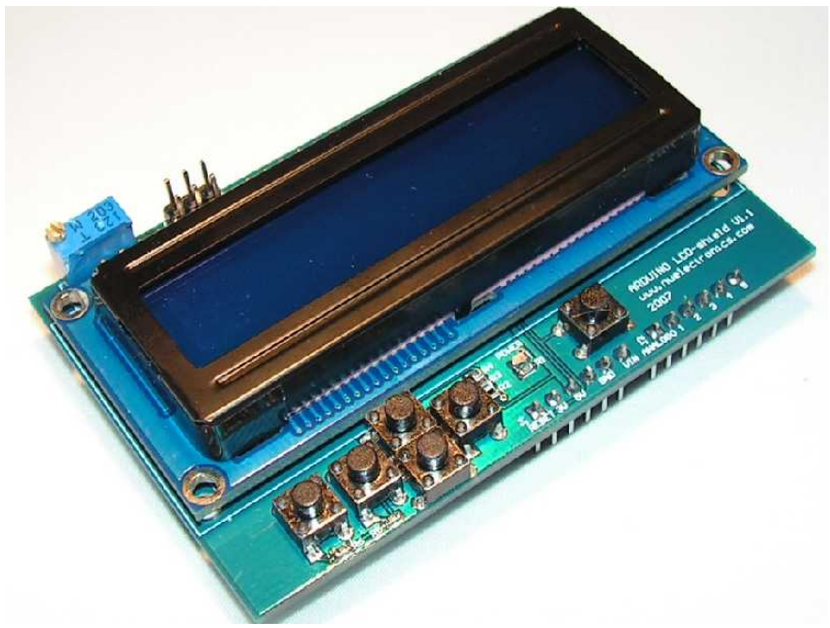


Motor Shield
by http://www.ladyada.net/



XBee Shield
by http://www.libelium.com/

XPort Shield
by http://www.ladyada.net/



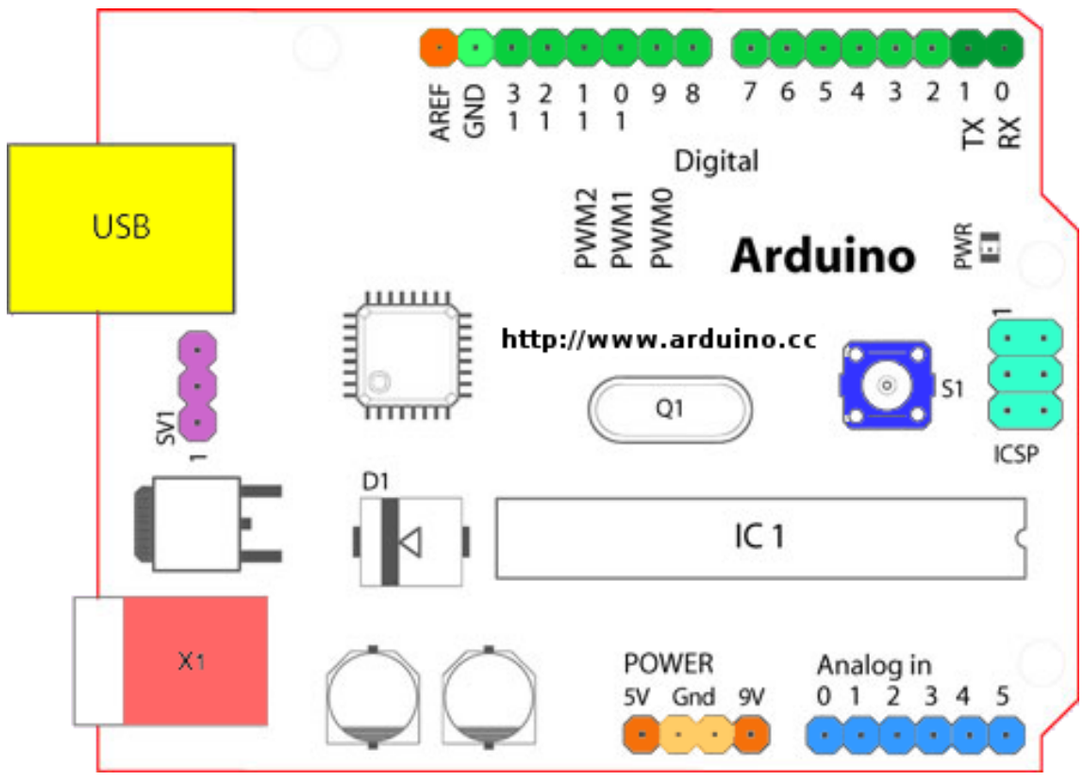
LCD Shield by http://www.nuelectronics.com

Arduino NG Diec

IC5
IN OUT
MC33269ST-5.0T3
IC4
MC33269D-5.0
VI VO
ADJ
VIN
VCC
D1
C6 100u
C5 100n
C7
GND
PWRIN
POWER
RESET
VCC30
VIN
+5V
L
R7 1k
C9 100n
AREF
SCK
MISO
MOSI
SS
J3
J2
J1
F1 500mA
USBVCC
C4 100n
C8 100n
VCC30
X4
USB
D-
D+
C10 100n
U4
VCC5I
VCCIO
VCC30
USBDM
USBDP
RESET#
OSCI
OSCO
GPIO1
GPIO0
GND1
GND2
GND3
TEST
AGND
VCC3I-NC
GPIO3
SLEEP#
FT232RL
R9 1k
R10 1k
TX
RX
R8 1k
PWR
PWR_SEL
USBVCC
VCC
CE
ATMEGA8
M8RXD R11 1k
M8TXD R12 1k

Hardware

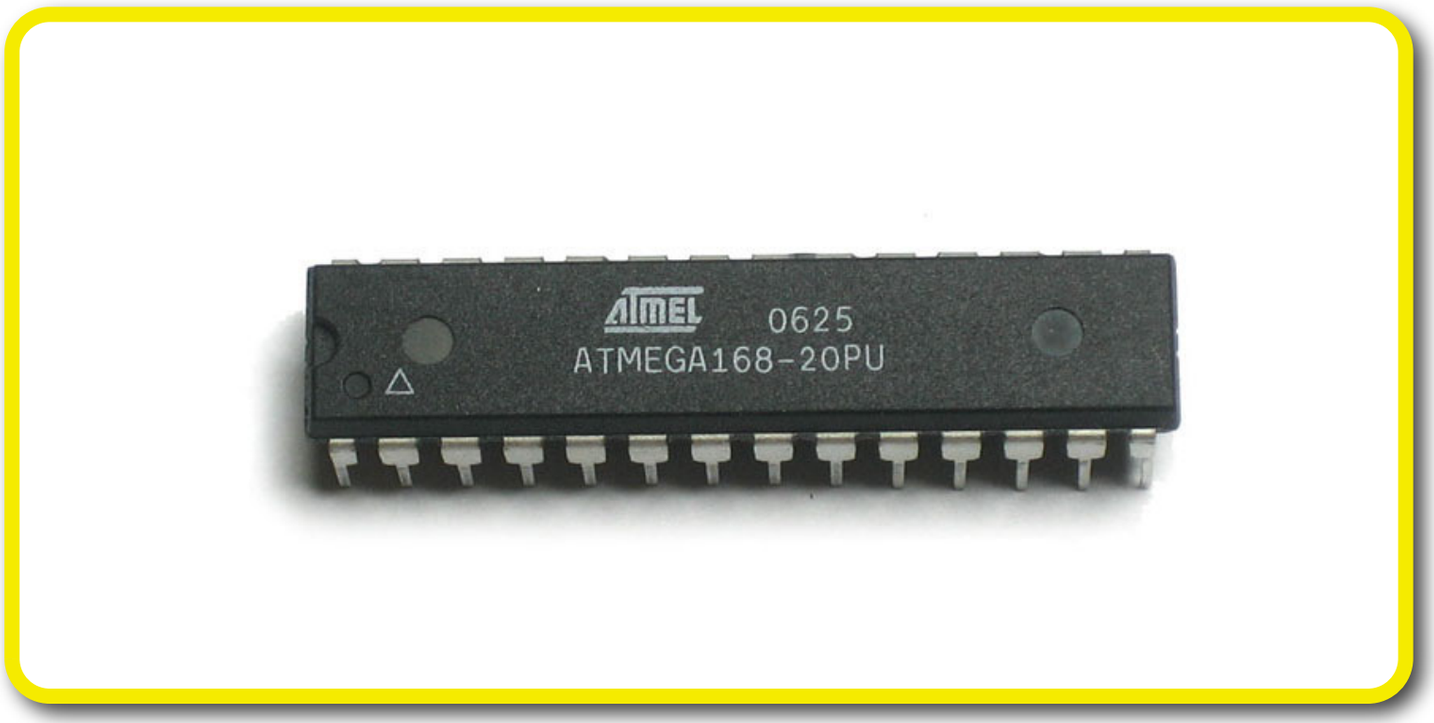
- Microcontroller ATMEL ATmega168
- Alimentazione: 6-20V (consigliati 7-12V) o via USB
- Ingressi/uscite digitali: 14 (6 output PWM)
- Ingressi analogici: 6
- Porta seriale TTL 5V
- Programmazione via USB
- Dimensioni: 7 x 5.5 cm



LAB Open Media Center
SARDEGNA DISTRICT

“Software”

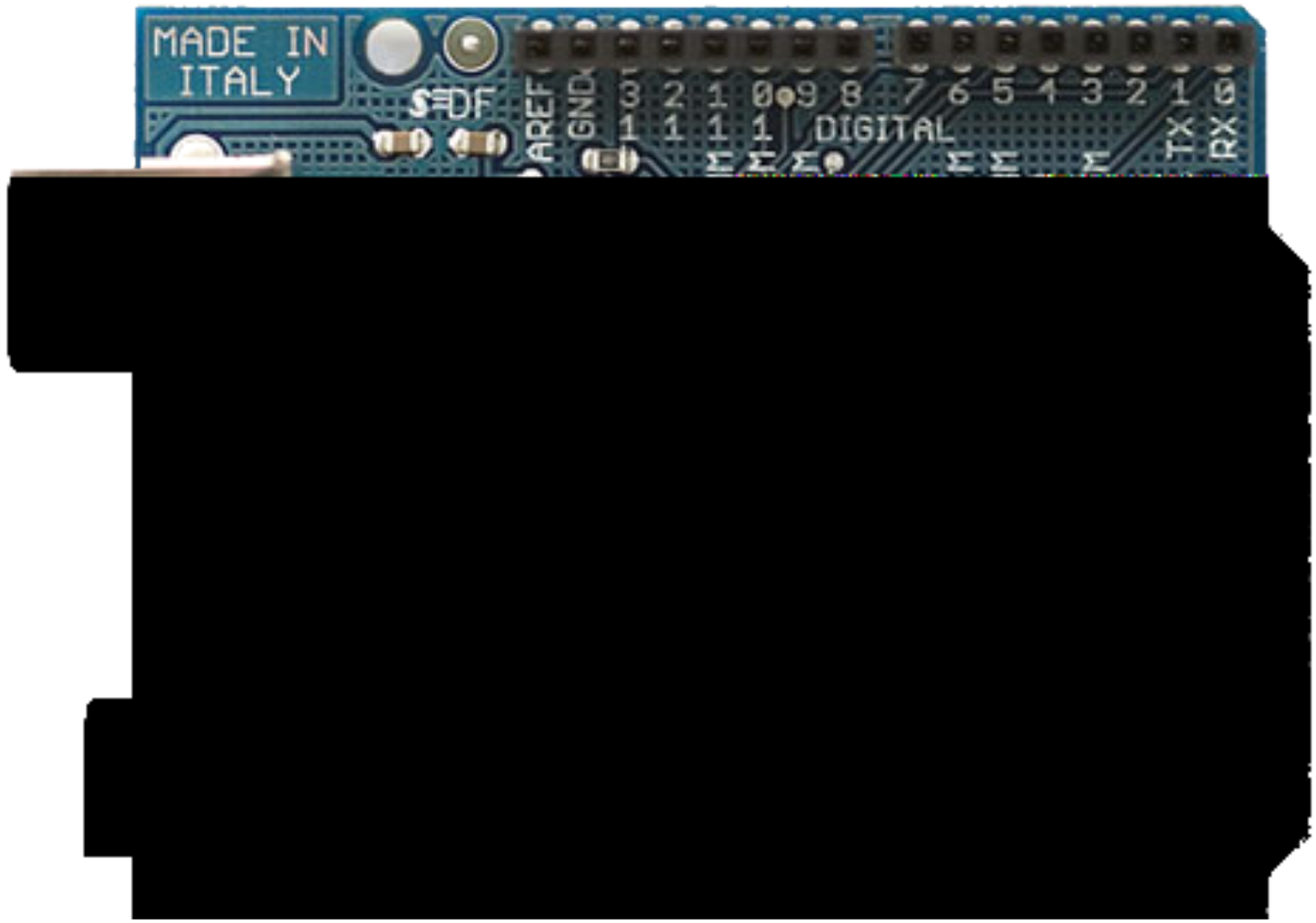
- Clock: 16Mhz
- Memoria flash: 16KB (di cui 14KB per i programmi utente)
- RAM: 1KB
- EEPROM: 512B



LAB Open MediaCenter

SARDEGNA DISTRICT

Arduino Diecimila

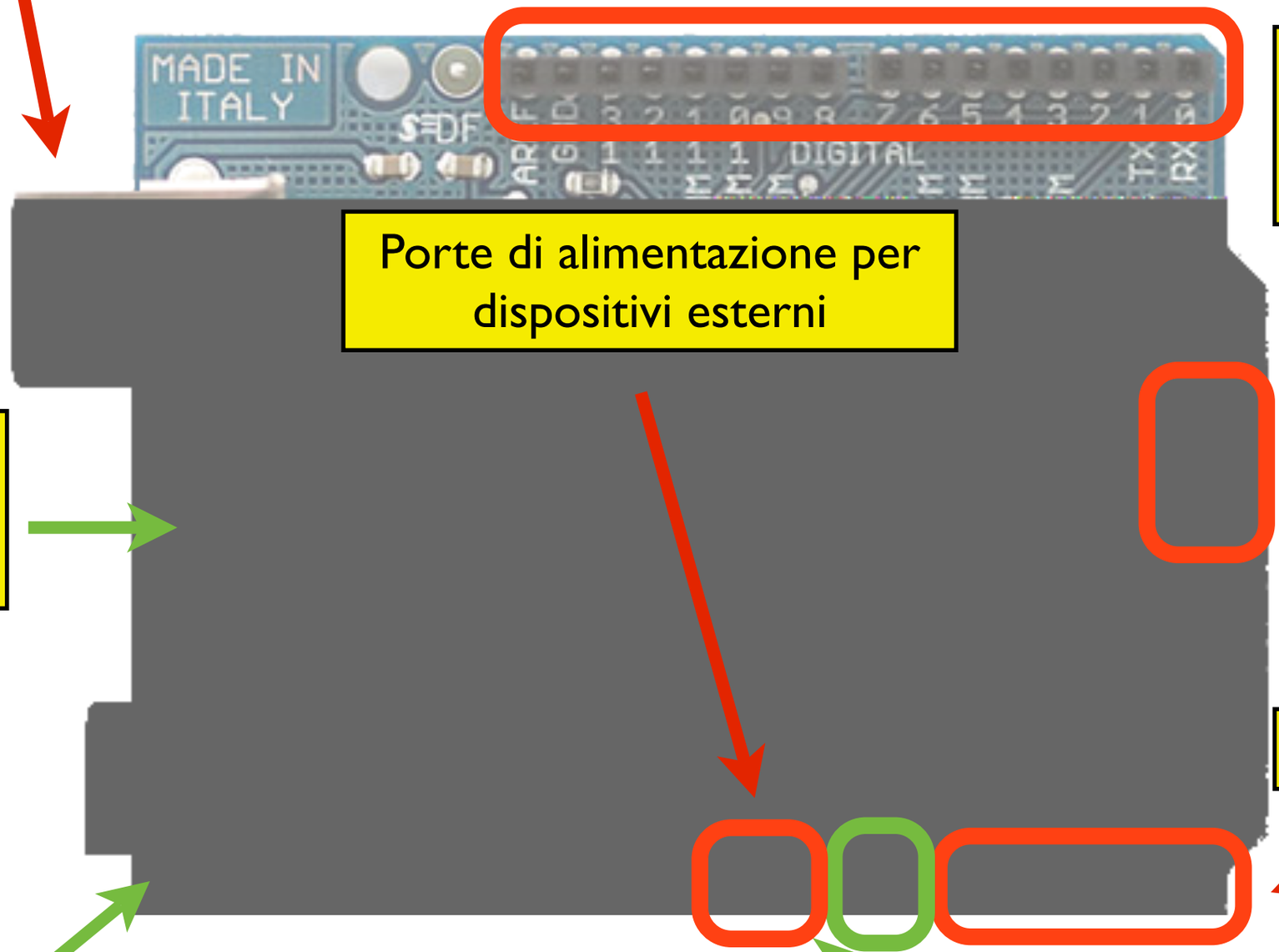


LAB Open Media Center
SARDEGNA DISTRICT

Interfacce

Porta USB

Input/Output digitali



Interfaccia di programmazione microcontrollore

Porte di alimentazione per dispositivi esterni

Selettore alimentazione (USB - esterna)

Ingressi analogici

Connettore di alimentazione

LAB Open Media Center
SARDEGNA DISTRICT

Interfacce

- Porta USB:
  - consente la comunicazione con il PC, sia durante la programmazione (caricamento nuovi sketch) che l'esecuzione dei programmi. La porta USB provvede anche ad alimentare il circuito.
- Connettori di alimentazione:
  - consentono di alimentare Arduino quando non connesso al PC. È necessaria una tensione continua compresa tra 7 e 12V
- Selettore alimentazione:
  - seleziona l'alimentazione via USB o attraverso alimentatore esterno.

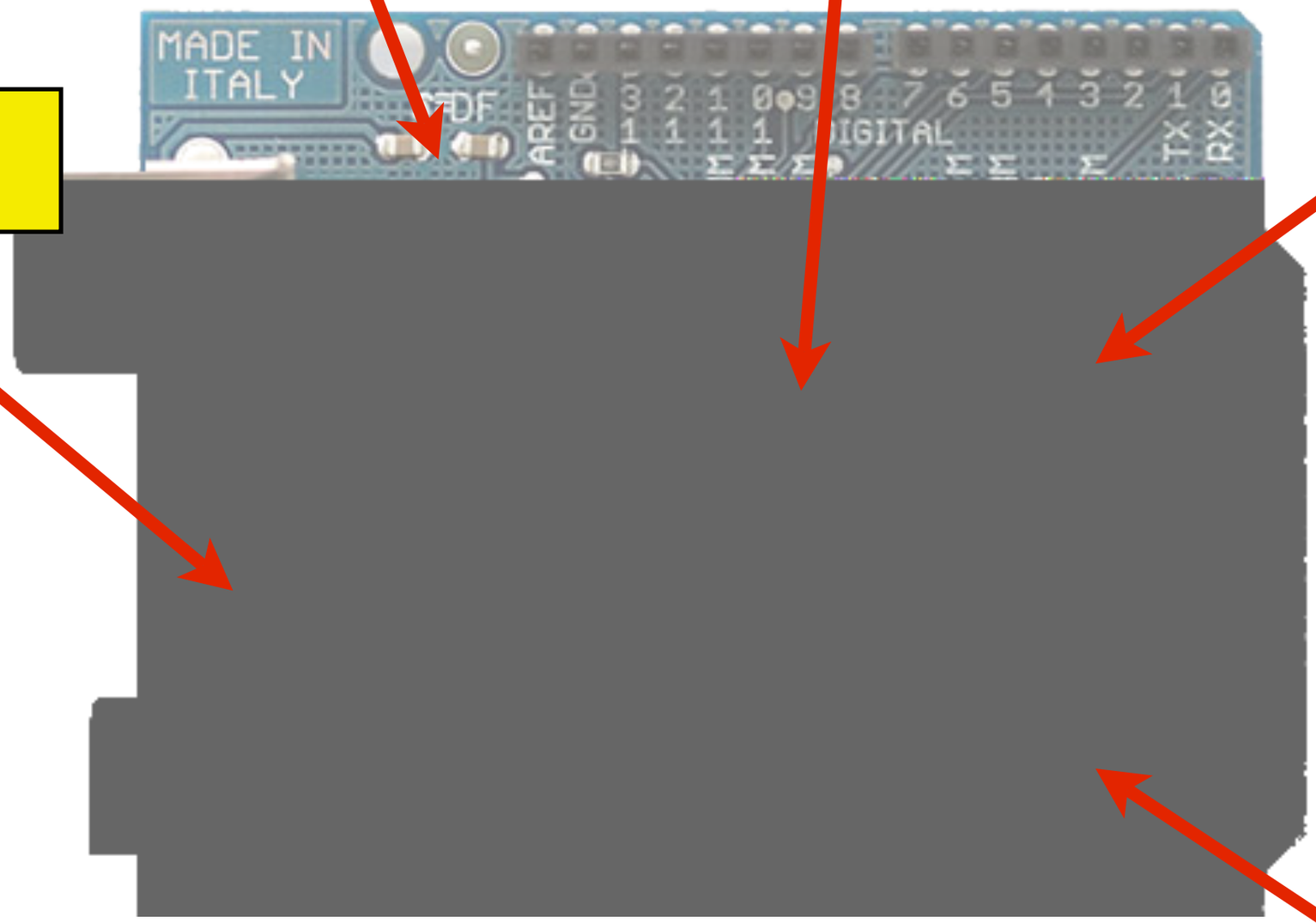
Componenti

Driver USB

Oscillatore

Reset

Regolatore di tensione



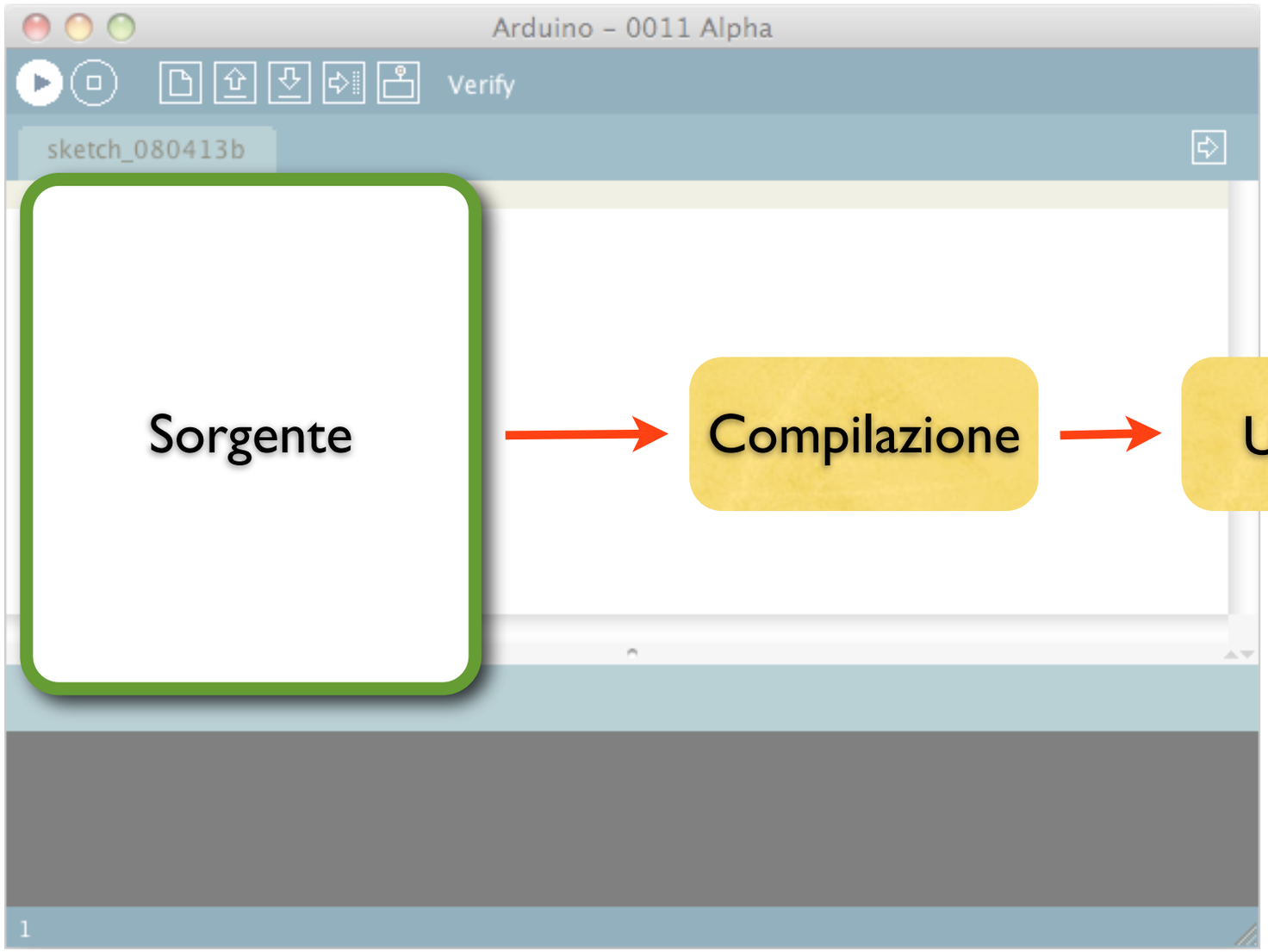
Microcontroller

LAB Open Media Center
SARDEGNA DISTRICT

Componenti

- Microcontroller:
  - esegue i programmi, acquisisce dati in ingresso e invia i dati in uscita.
- Driver USB:
  - realizza l'interfaccia tra la seriale dell'ATmega e la posta USB del PC
- Oscillatore:
  - è "l'orologio interno" di Arduino, con il quale il microcontrollore esegue i programma e gestisce i flussi dati digitali
- Regolatore di tensione:
  - provvede a mantenere costante la tensione di alimentazione

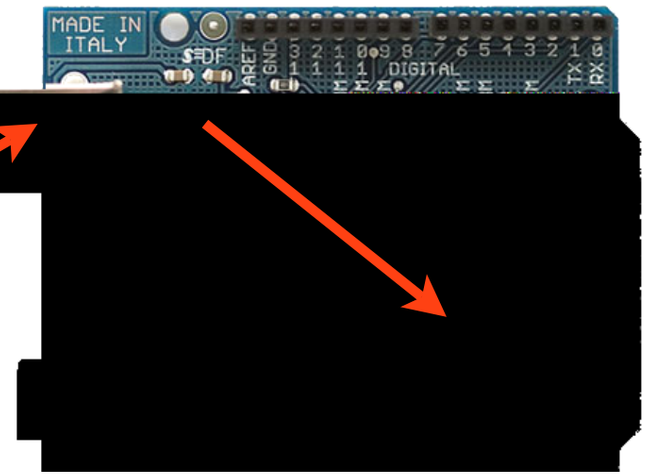
Processo di sviluppo



Sorgente

Compilazione

Upload

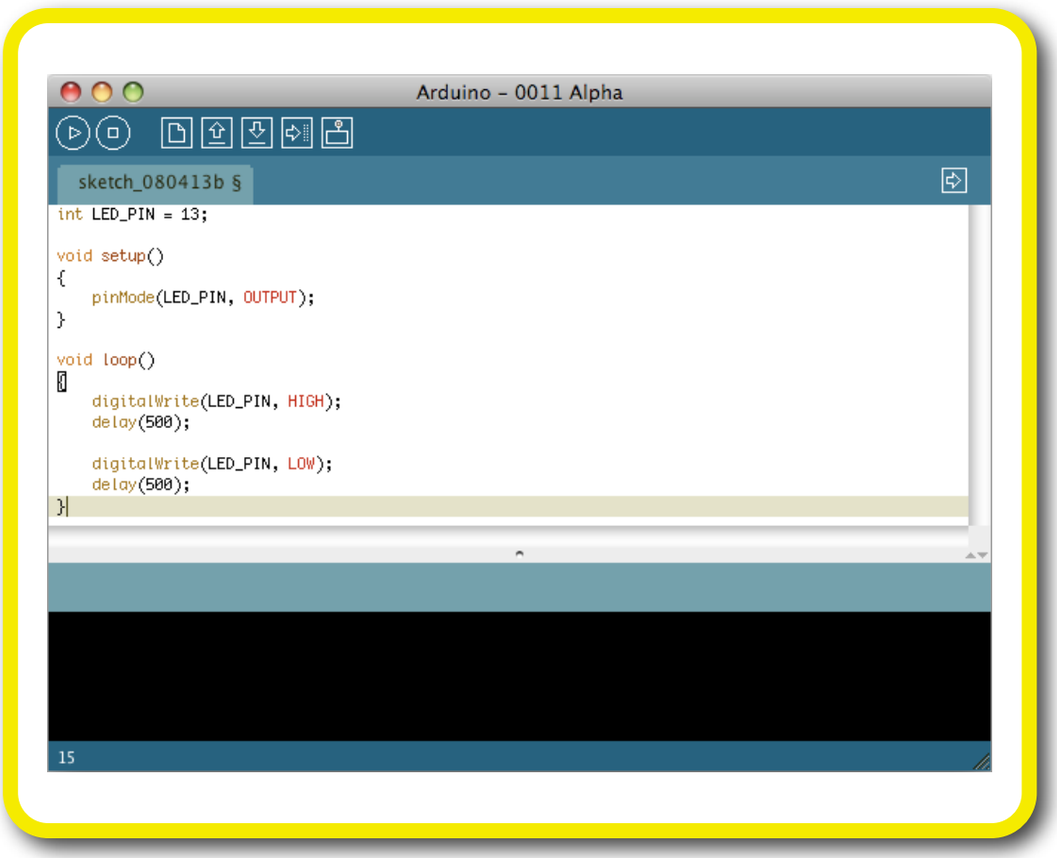


Ambiente di sviluppo

- Per scrivere programmi (chiamati sketch) per Arduino è necessario utilizzare un software open source scaricabile gratuitamente all'indirizzo:

  http://www.arduino.cc

- L'ambiente di sviluppo è disponibile per Windows, Mac OS X e Linux



LAB Open Media Center
SARDEGNA DISTRICT

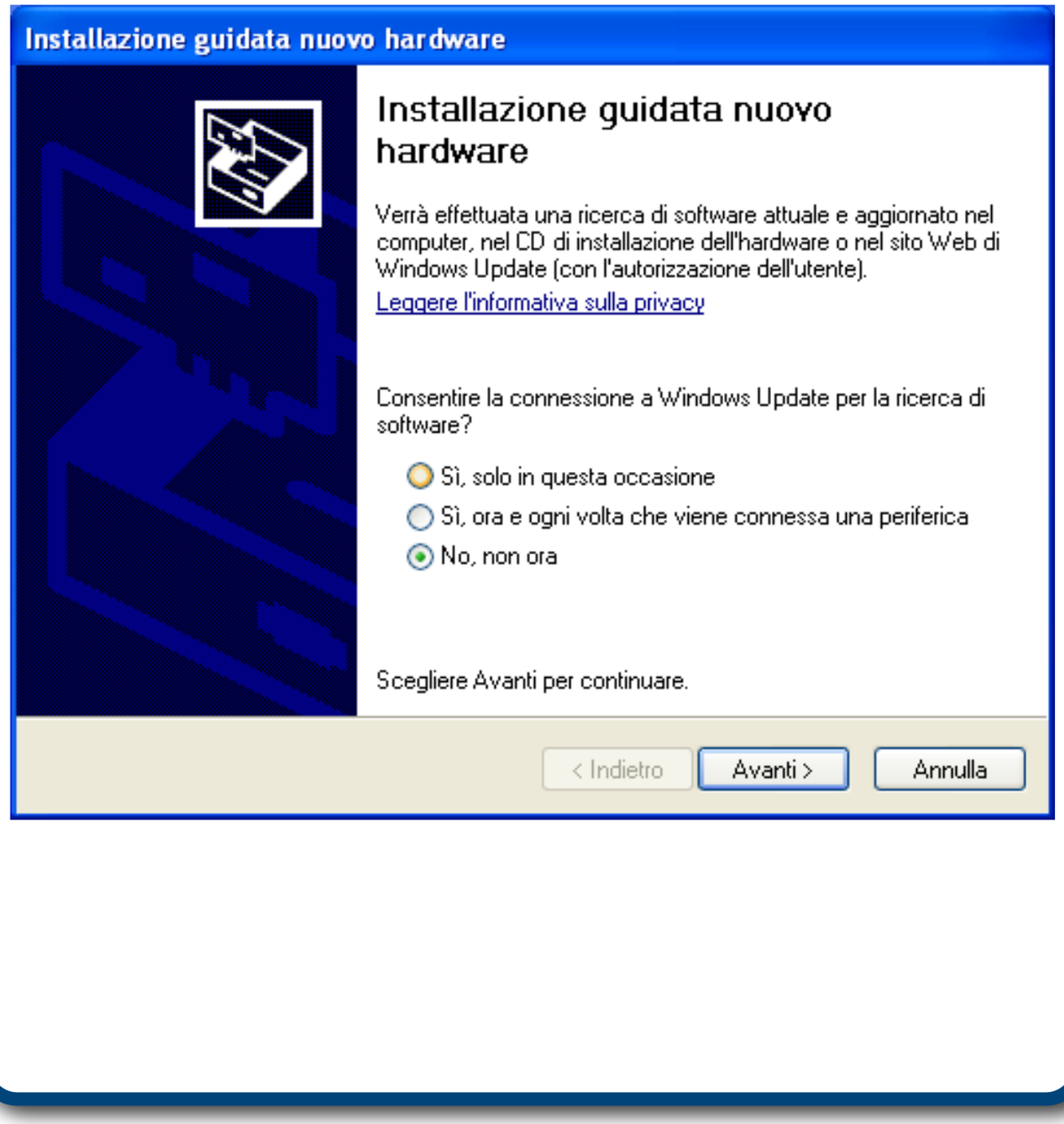
Installazione

- Windows
  - Scaricare il file **http://www.arduino.cc/files/arduino-0012-win.zip** e decomprimerlo in una cartella
  - L'installazione del driver per l'interfaccia USB-FTDI per Arduino è richiesta automaticamente dal sistema operativo connettendo Arduino alla porta USB: indicare il driver contenuto nella cartella **drivers/FTDI USB Drivers**
- Mac OS X (Tiger e Leopard)
  - Scaricare il file **http://www.arduino.cc/files/arduino-0012-mac.zip** e decomprimerlo in una cartella
  - Installare il driver per l'interfaccia USB-FTDI per Arduino, contenuto nella cartella drivers (**FTDIUSBSerialDriver_v2_1_9.dmg** per PowerPC e

LAB Open MediaCenter
SARDEGNA DISTRICT

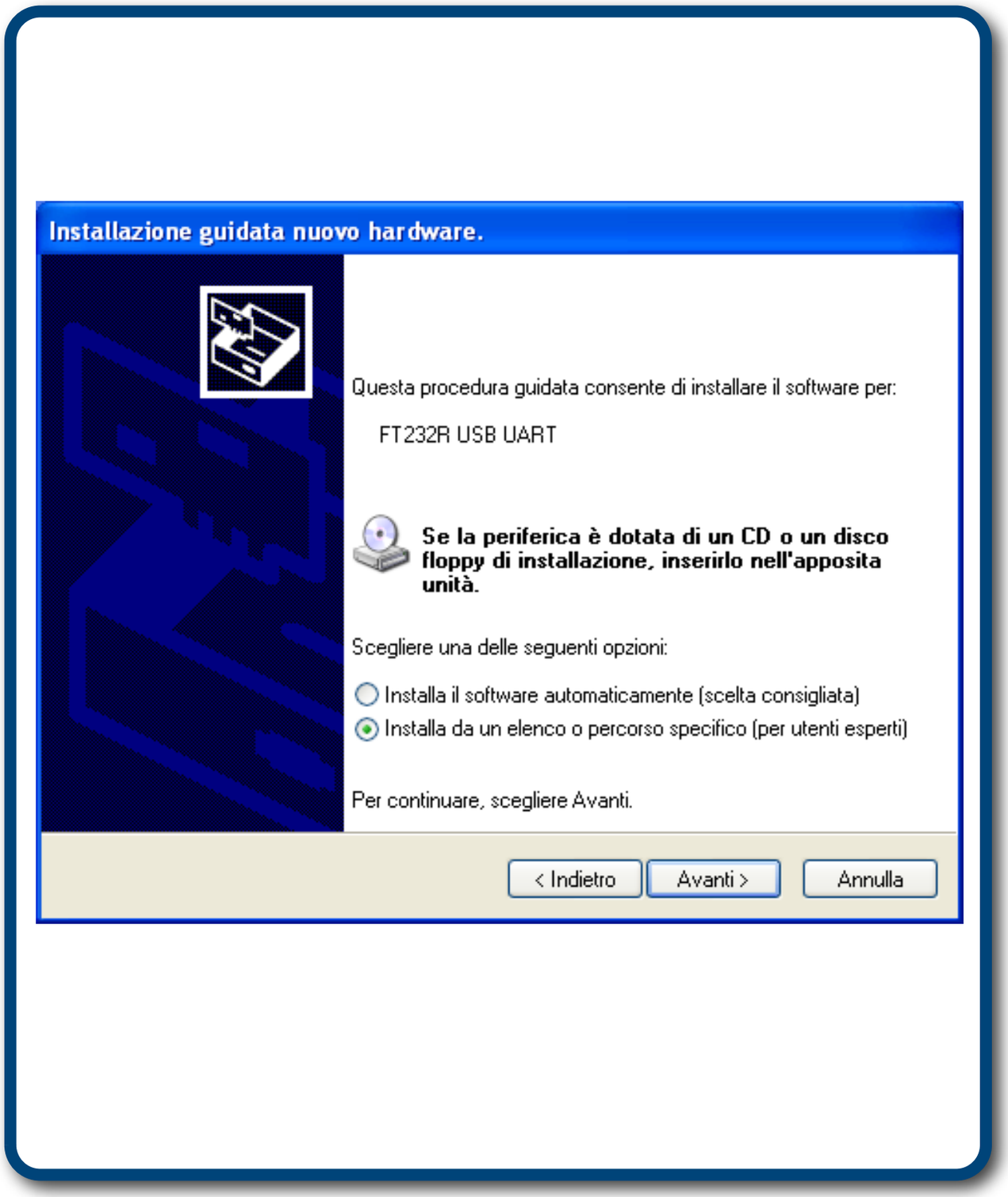
Installazione

- Connettere la scheda ad una USB
- Il sistema avvisa che è stato trovato un nuovo hardware
- Non è necessario effettuare la connessione al servizio Windows Update



Installazione

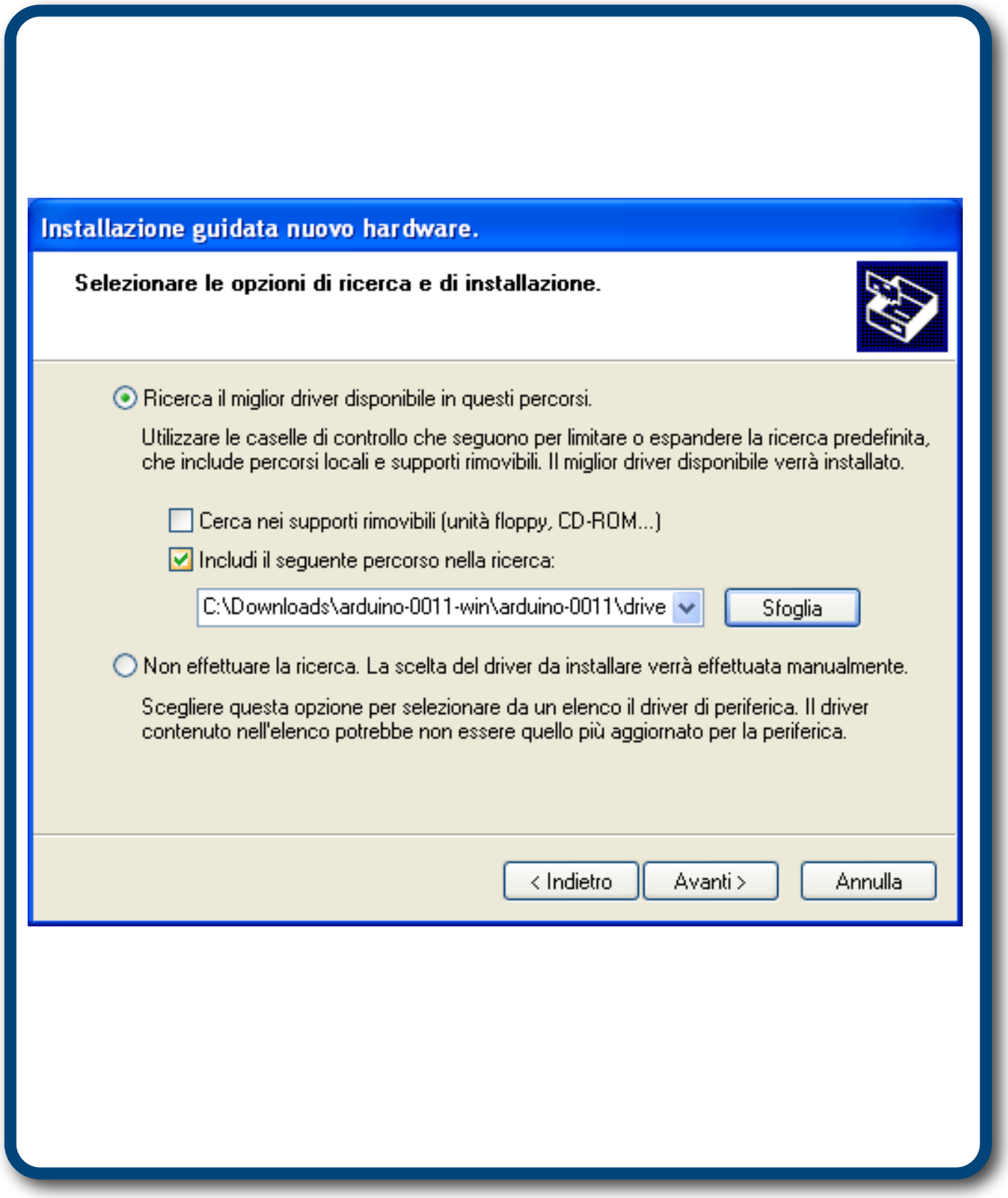
- Il primo driver da installare è per il device **FT232R USB UART**
- Selezionare l'installazione da percorso specifico (non selezionare la modalità automatica)



LAB Open MediaCenter
SARDEGNA DISTRICT

Installazione

- Indicare il percorso di ricerca dei driver nella cartella **driver** del folder dove è stato decompresso l'archivio di Arduino

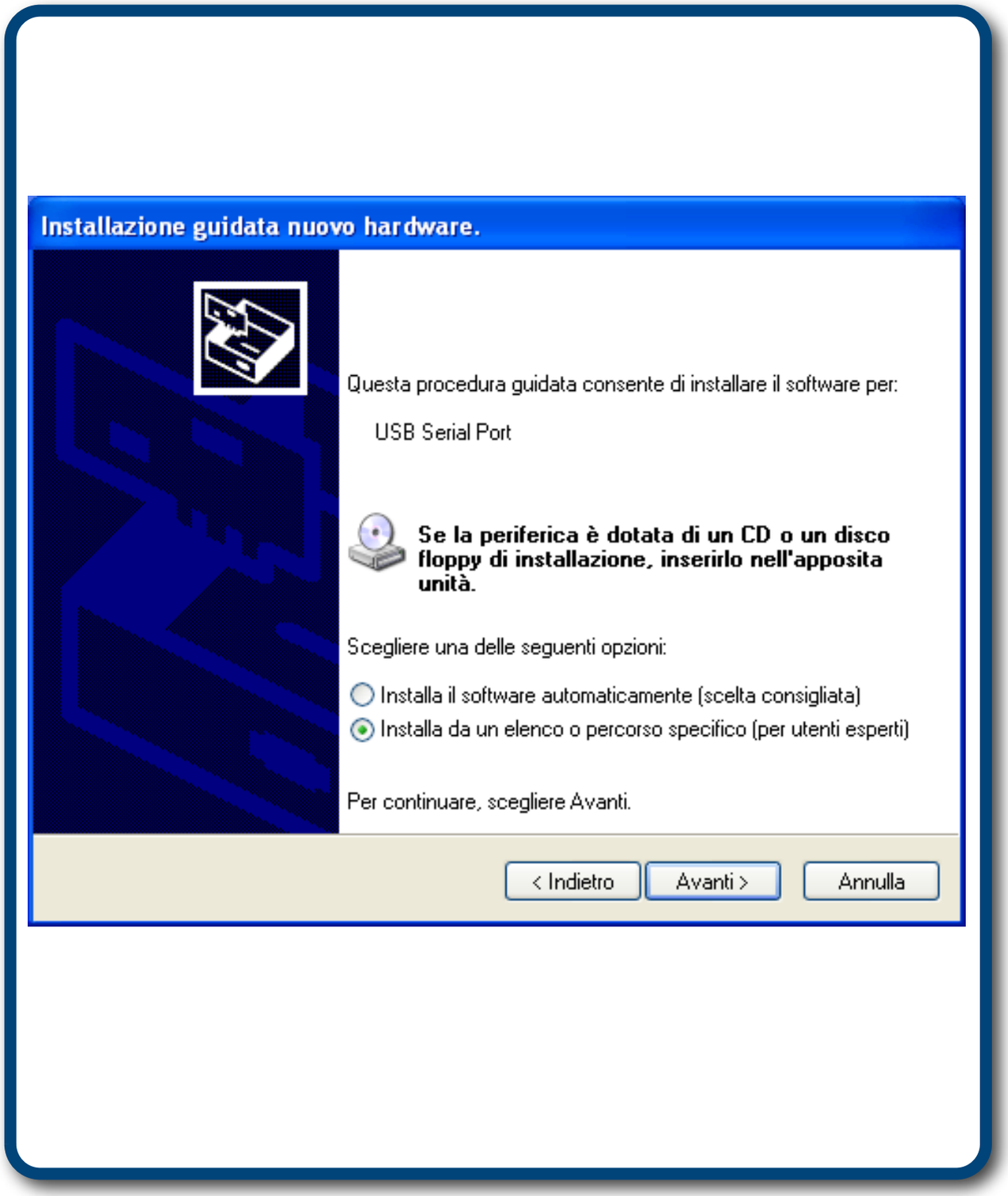


Installazione

- Dopo alcuni secondi, il primo driver è installato

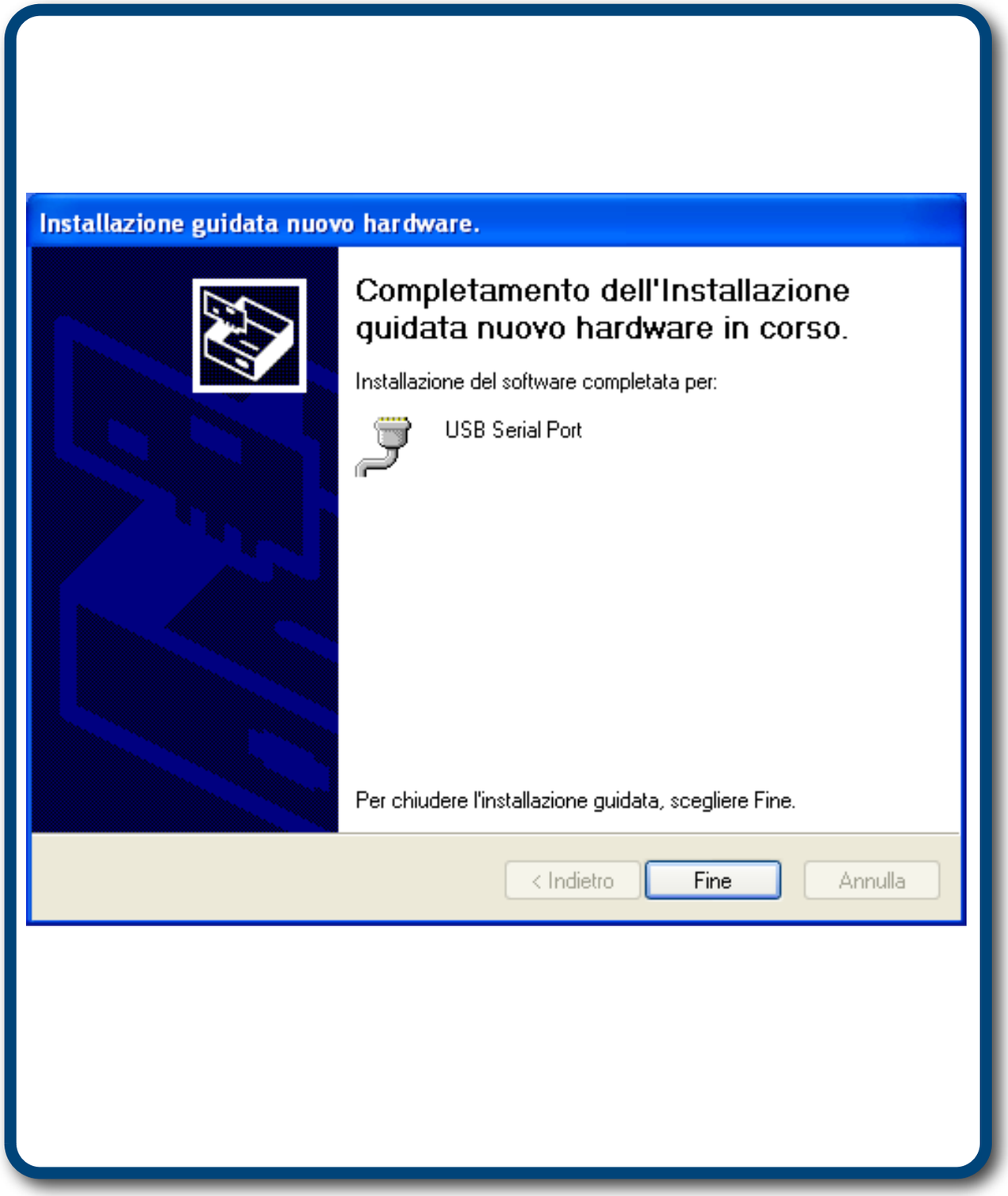
Installazione

- È ora necessario installare il driver per il dispositivo **USB Serial Port**
- Valgono le stesse selezioni viste nelle slide precedenti



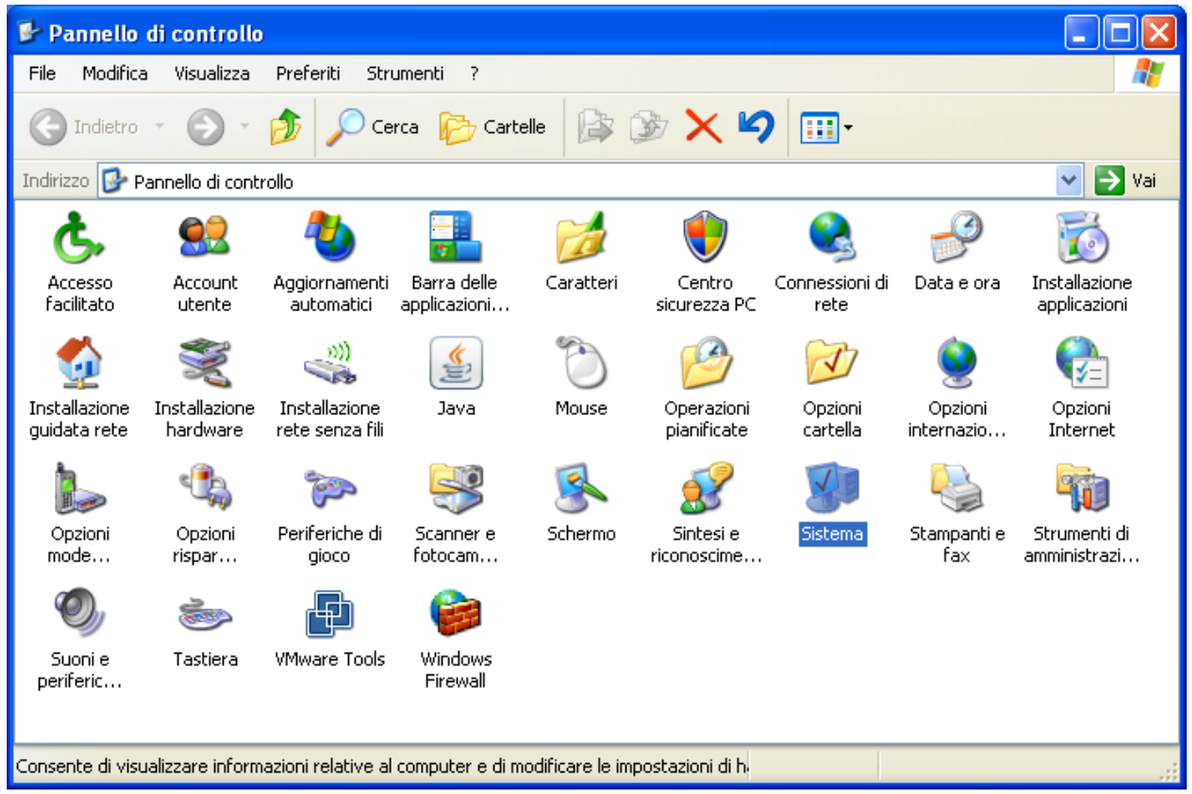
Installazione

- Dopo alcuni secondi, il secondo driver è installato



Installazione

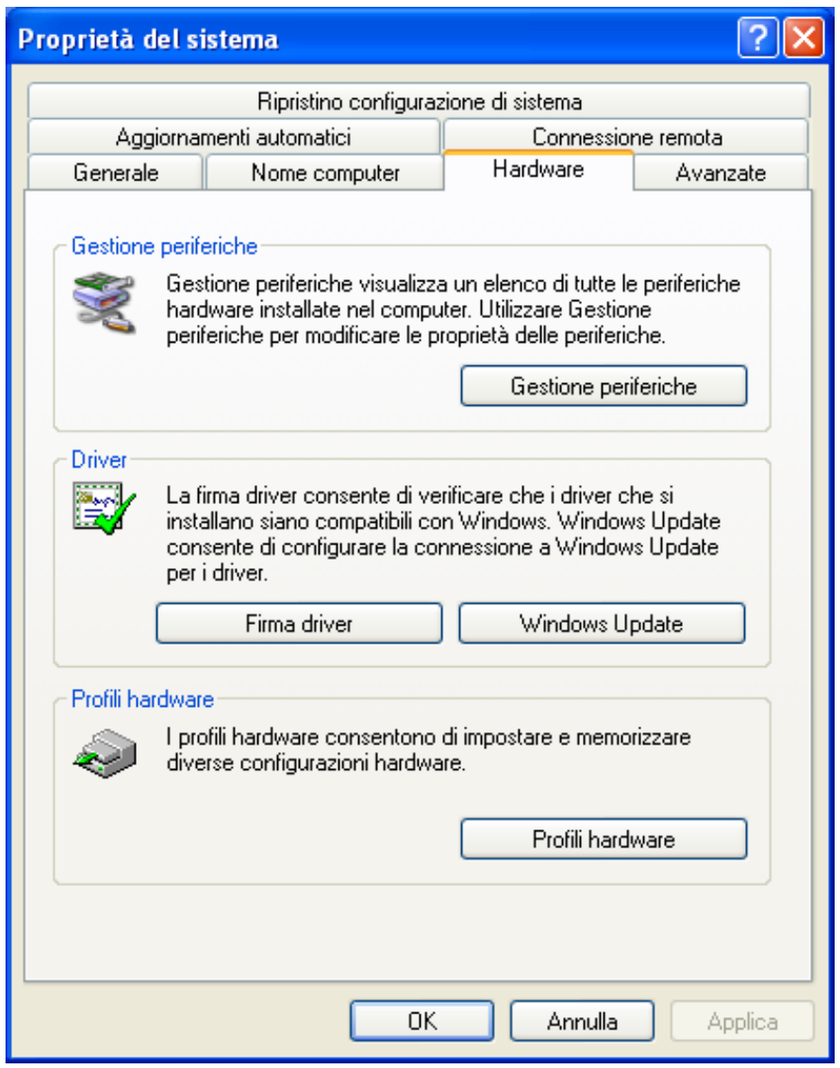
- È ora necessario individuare la porta seriale virtuale che è stata assegnata al driver FTDI associato ad Arduino
- Aprire il pannello di controllo e selezionare l'icona **Sistema**



LAB Open MediaCenter
SARDEGNA DISTRICT

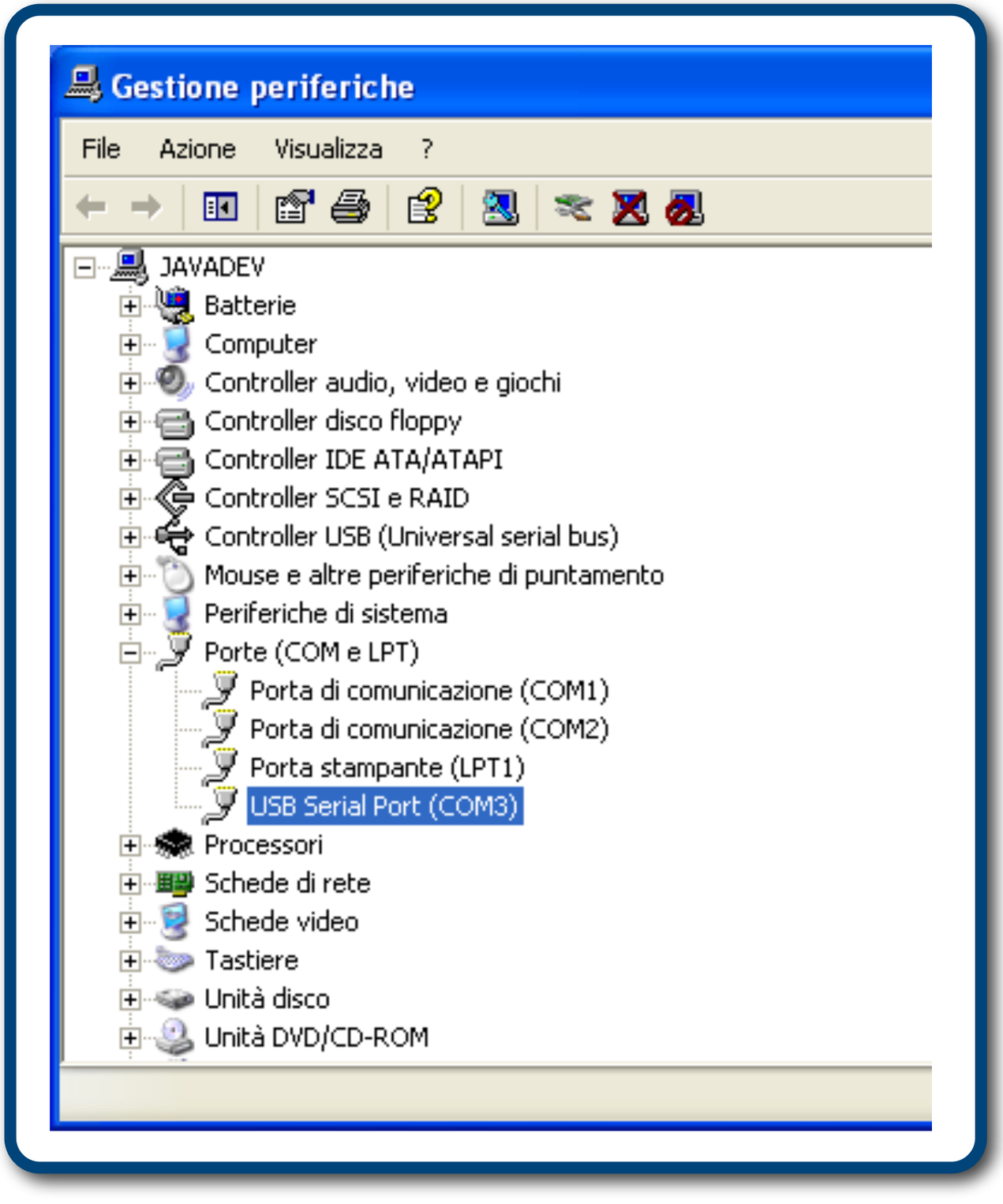
Installazione

- Selezionare il pannello **Hardware** e il pulsante **Gestione periferiche**



Installazione

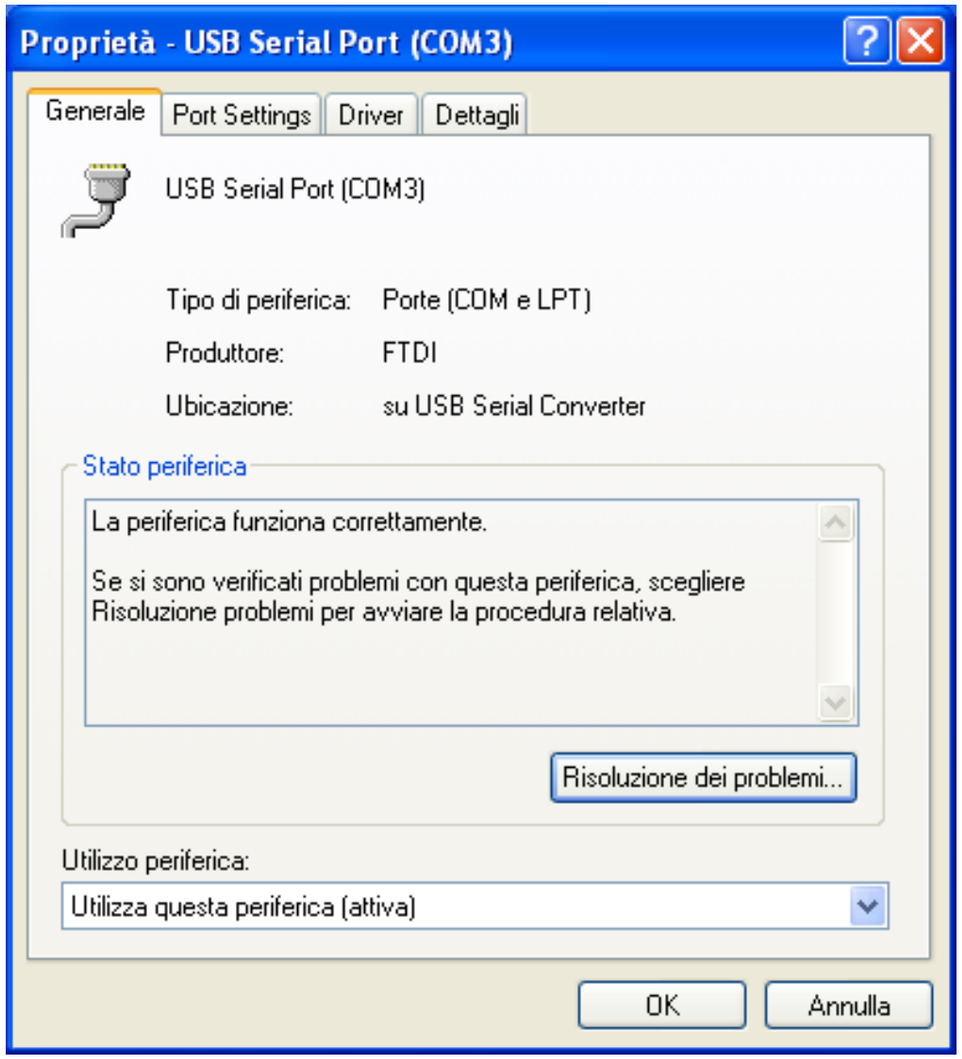
- Su Gestione periferiche, selezionare il gruppo Porte (COM e LPT)
- Tra le porte seriali elencate, selezionare **USB Serial Port**: tra parentesi è indicato il nome assegnato alla porta



LAB Open MediaCenter
SARDEGNA DISTRICT

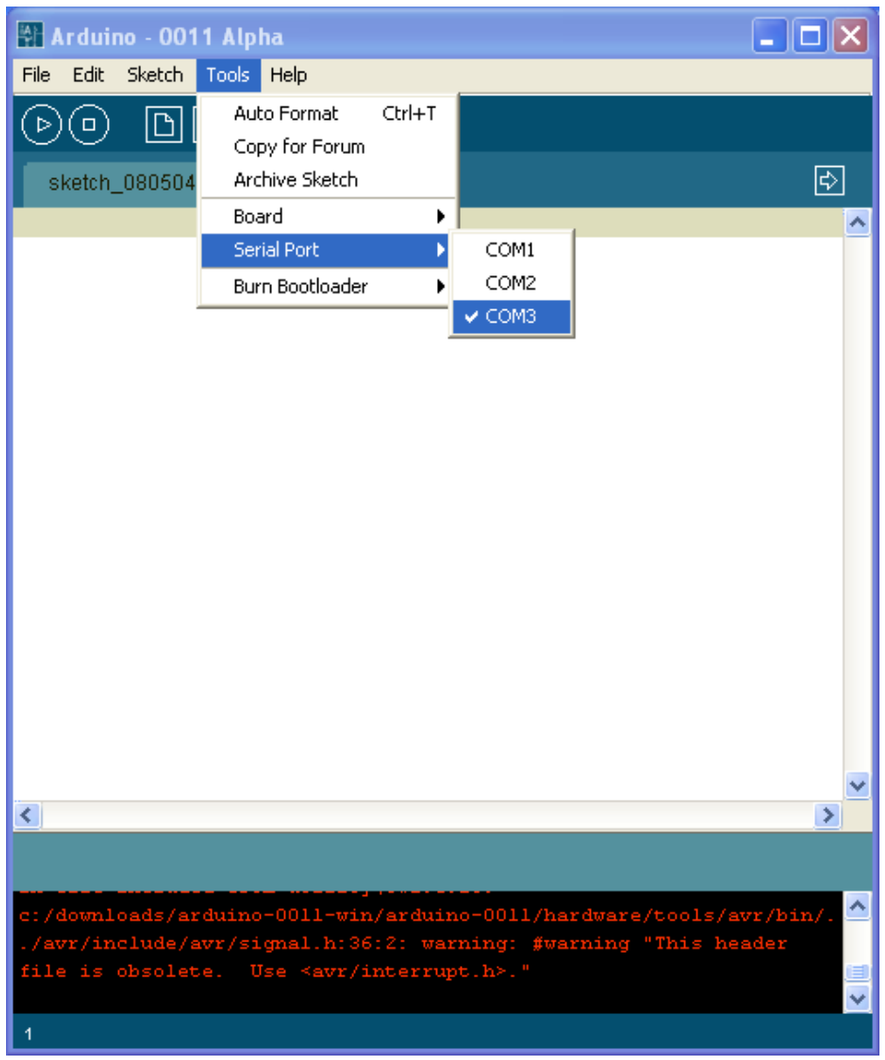
Installazione

- Verificare che si tratti di un dispositivo del produttore **FTDI**



Configurazione

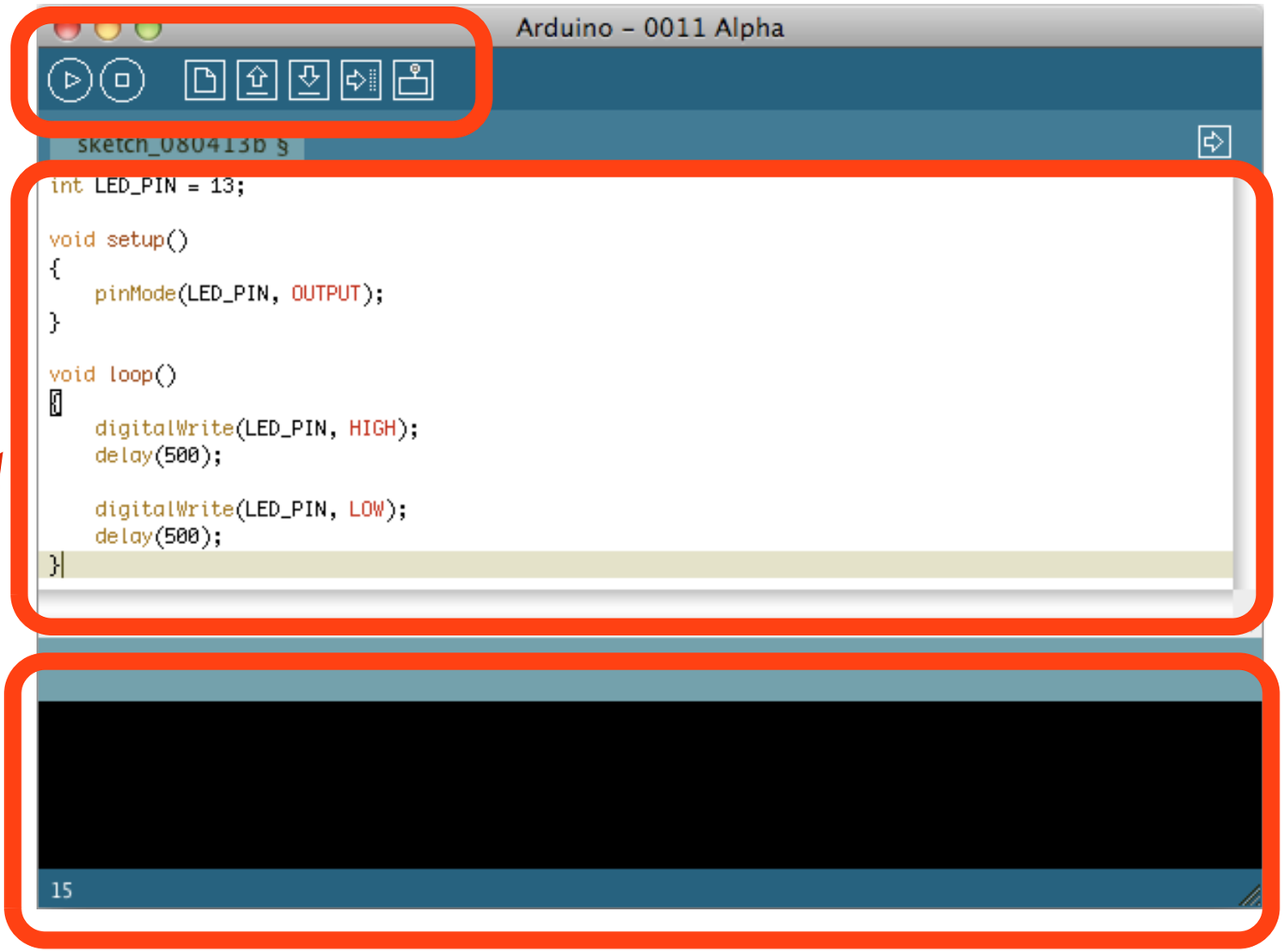
- Avviando l'ambiente di sviluppo di Arduino, è ora possibile indicare la porta seriale associata alla scheda attraverso il menu **Serial Port** del gruppo **Tools**



LAB Open MediaCenter
SARDEGNA DISTRICT

IDE

Toolbar



Editor

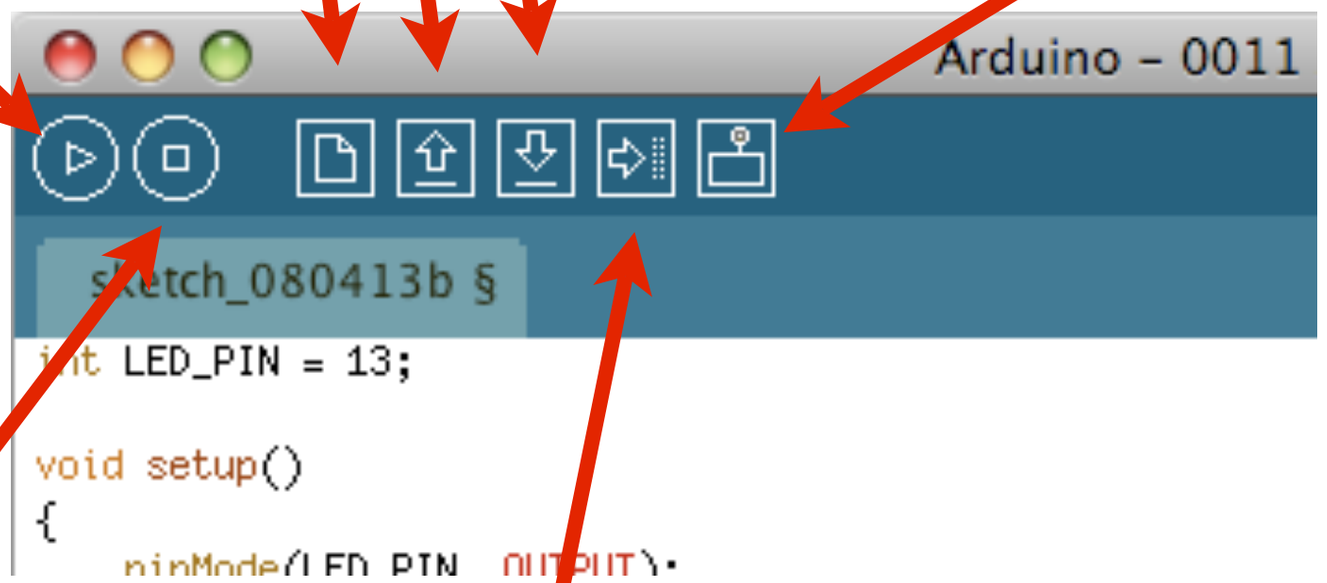
Console

IDE Toolbar

Gestione file: nuovo, apri, salva

Attivazione console seriale

Compilazione



Chiusura console seriale

Compilazione e caricamento programma su Arduino

LAB Open Media Center
SARDEGNA DISTRICT

Hello Arduino!

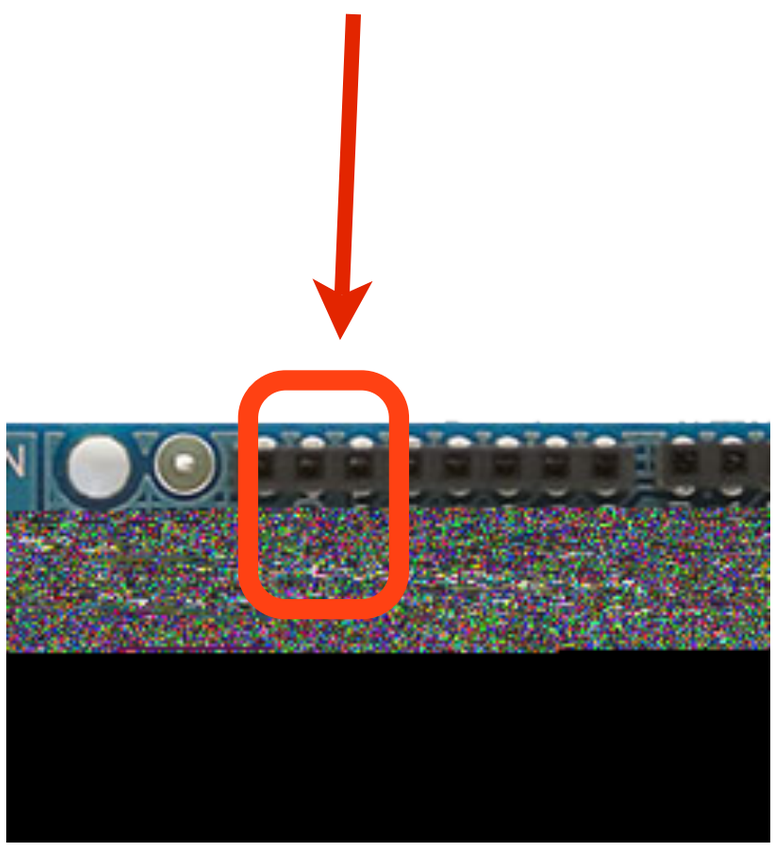
- Nella programmazione di sistemi embedded, il programma più semplice che è possibile scrivere svolga una sola funzione essenziale: accende e spegne un LED ad intervalli regolari

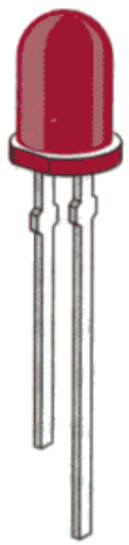
- Pur nella sua semplicità, questo esempio consente di conoscere il processo di sviluppo e di cablaggio di un circuito!

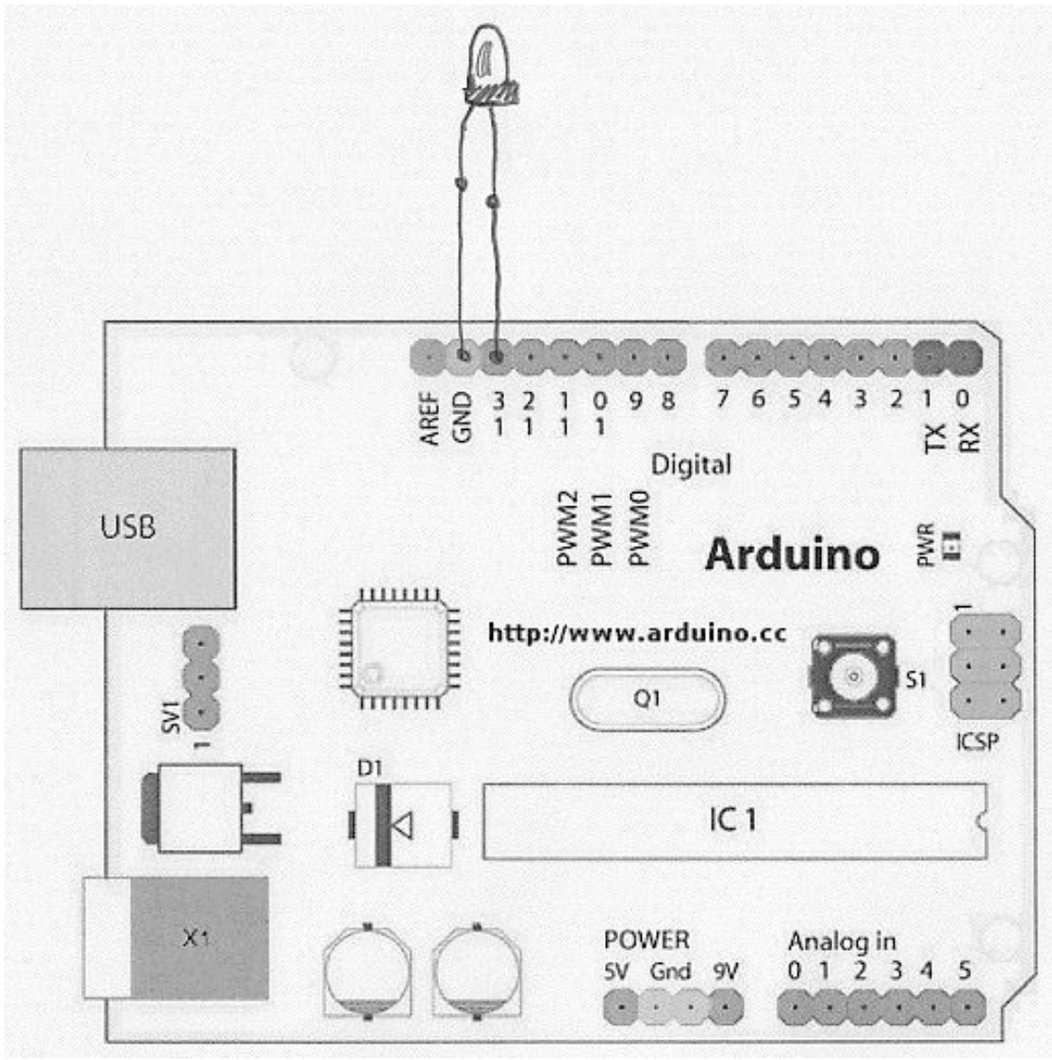
LAB Open Media Center
SARDEGNA DISTRICT

Hello Arduino!

PIN 13 e
PIN GND (ground, “massa”)



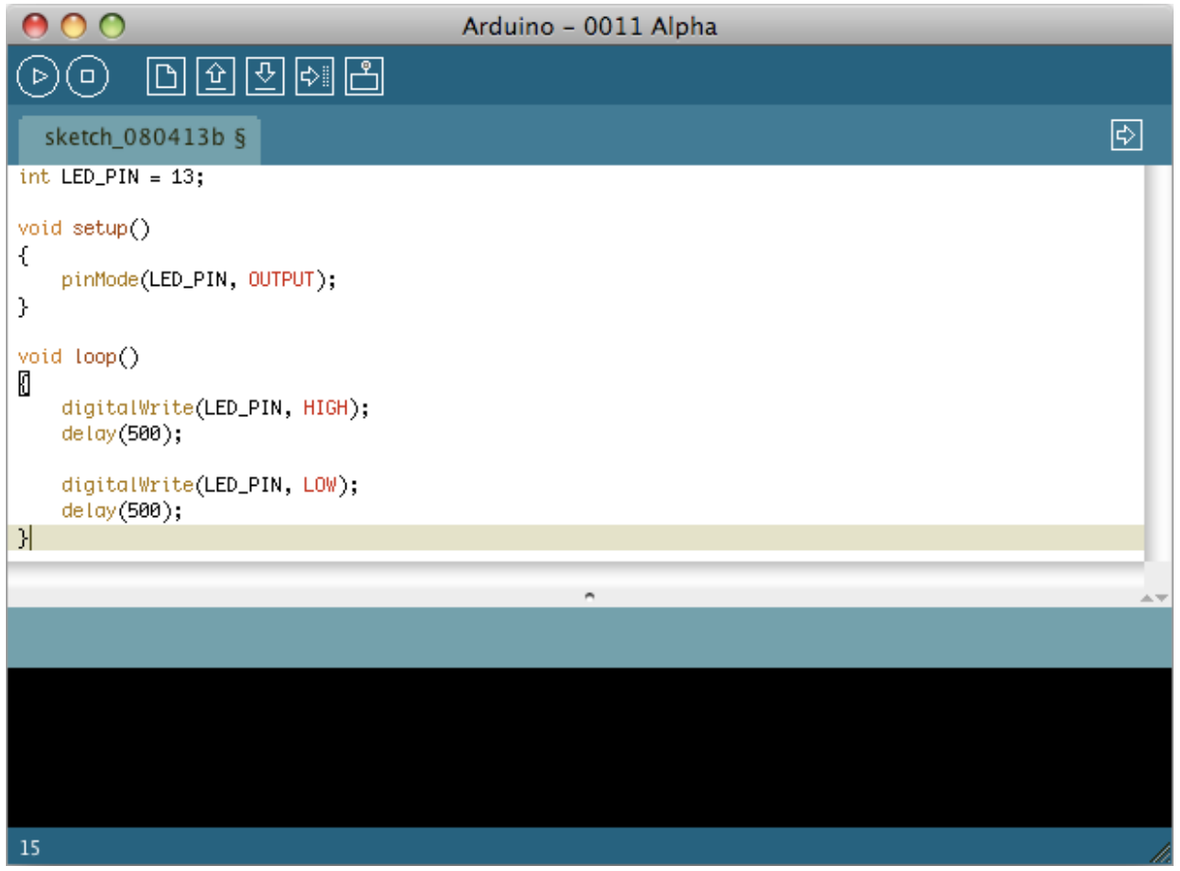




LAB Open MediaCenter
SARDEGNA DISTRICT

Hello Arduino!

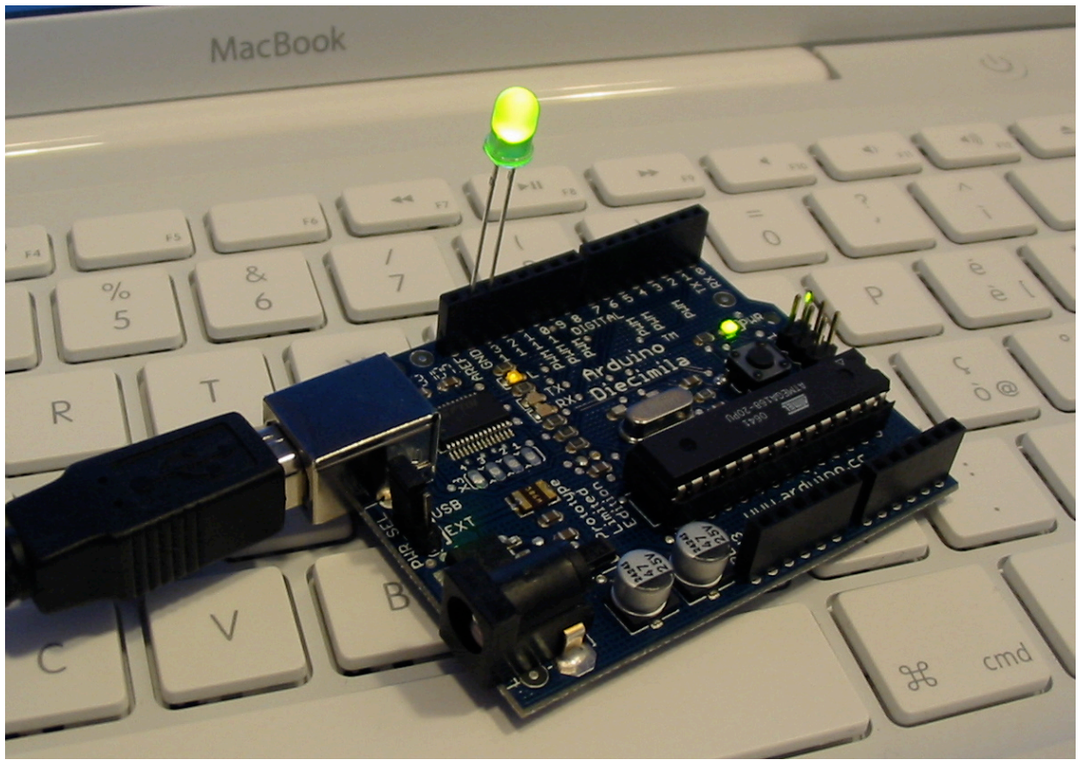
```
int LED_PIN = 13;

void setup()
{
    pinMode(LED_PIN, OUTPUT);
}

void loop()
{
    digitalWrite(LED_PIN, HIGH);
    delay(500);

    digitalWrite(LED_PIN, LOW);
    delay(500);
}
```



LAB Open Media Center
SARDEGNA DISTRICT

Hello Arduino!

```
int LED_PIN = 13;

void setup()
{
    pinMode(LED_PIN, OUTPUT);
}

void loop()
{
    digitalWrite(LED_PIN, HIGH);
    delay(500);

    digitalWrite(LED_PIN, LOW);
    delay(500);
}
```



Programmazione

Linguaggio e librerie

SARDEGNA DISTRICT

LAB Open MediaCenter

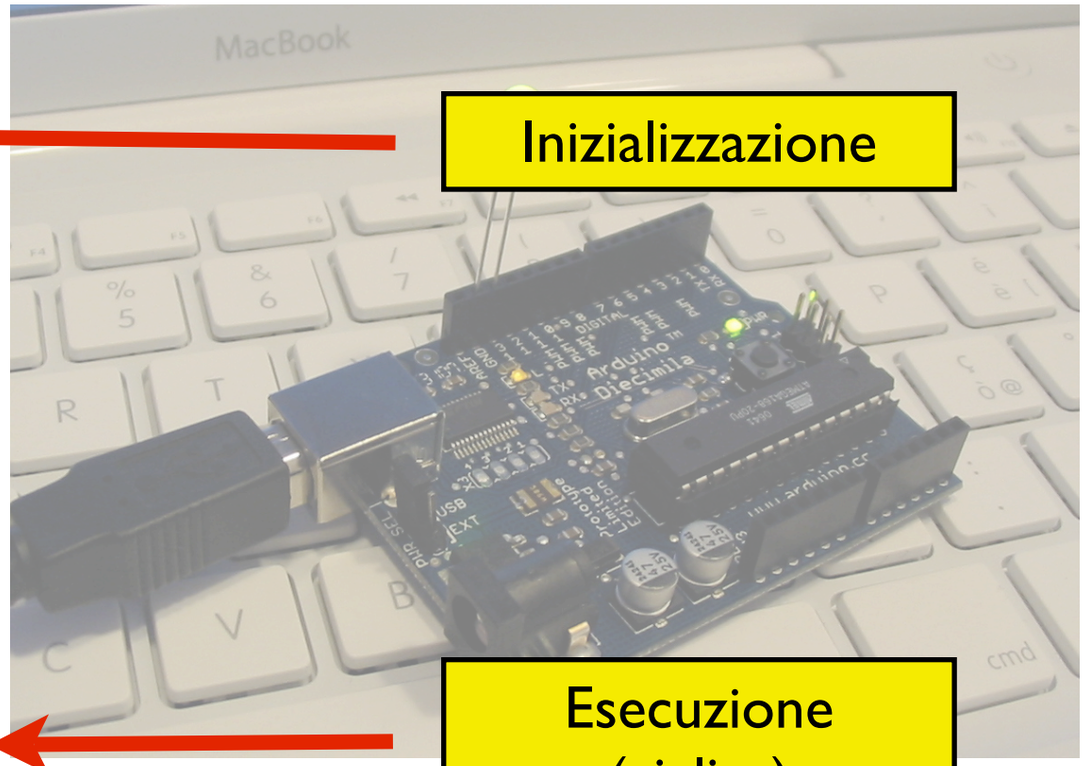
CRS4

Struttura di un sketch

```
int LED_PIN = 13;

void setup()
{
    pinMode(LED_PIN, OUTPUT);
}

void loop()
{
    digitalWrite(LED_PIN, HIGH);
    delay(500);

    digitalWrite(LED_PIN, LOW);
    delay(500);
}
```

Variabili

Inizializzazione



Esecuzione
(ciclica)

LAB Open Media Center
SARDEGNA DISTRICT

Come in C...

- Variabili: byte, int, unsigned int, long, unsigned long, float, double, boolean (true/false), char, array
- Costanti:
  - Livelli logici: HIGH-LOW
  - Valori booleani: true-false
  - Stato dei PIN: INPUT-OUTPUT
- Operatori, funzioni (con o senza valore di ritorno), controllo di flusso

LAB Open Media Center
SARDEGNA DISTRICT

Funzioni

- Permettono di calcolare:
  - massimo e minimo
  - valore assoluto
  - estrazione della radice quadrata
  - elevazione a potenza
  - vincolo
  - range

```
int a = 0; int b = 16; int c =
-3;

a = constraint(b, 0, 10);
    // a = 10
a = constraint(b, 0, 20);
    // a = 16

a = map(b, 0, 20, 0, 10);
    // a = 8
```

Libreria standard

- Funzioni predefinite che si occupano di:
  - impostare l'hardware (I/O, memoria, porta di comunicazione)
  - leggere i dati in input
  - inviare dati in uscita
  - facilitare l'uso di funzionalità comuni (accesso a display, controllo di motori...)

```
//  I/O digitale
pinMode(pin, mode);
digitalWrite(pin, value);
int digitalRead(pin);

//  I/O analogico
int analogRead(pin);
analogWrite(pin, valore);

//  Temporizzazione
unsigned long millis();
delay(ms);
delayMicroseconds(us);
```

LAB Open Media Center
SARDEGNA DISTRICT

Libreria standard

- inviare treni di bit
- leggere la durata di un impulso in ingresso
- ...

```
// scrittura di treni di bit
shiftOut(dataPin, clockPin,
bitOrder, value)

// lettura di impulsi
unsigned long pulseIn
                    (pin, value)
unsigned long pulseIn
          (pin, value, timeout)
```

LAB Open Media Center
SARDEGNA DISTRICT

Librerie estese

- Ufficiali
  - EEPROM: reading and writing to "permanent" storage
  - LiquidCrystal: for controlling liquid crystal displays (LCDs)
  - Servo: for controlling servo motors
  - Stepper - for controlling stepper motors
  - SoftwareSerial: for serial communication on any digital pins
  - Wire - Two Wire Interface (TWI/I2C) per reti di sensori
  - Matrix: Basic LED Matrix display manipulation library
  - Sprite: Basic image sprite manipulation library for use in animations with an LED matrix

LAB Open Media Center
SARDEGNA DISTRICT

Librerie estese

- Community
  - DateTime: gestione data e ora via software
  - Firmata: comunicazione seriale con applicazioni su PC
  - GLCD: driver per LCD basati su chipset KS0108
  - LCD e LCD 4 bit: controller LCD (8 e 4 data lines)
  - LedControl: driver per LED matrix con MAX7221 /7219.
  - TextString: gestione stringhe
  - Metro e MsTimer: per la lo scheduling di azioni

LAB Open Media Center
SARDEGNA DISTRICT

Librerie estese

- Community
  - OneWire: driver per dispositivi compatibili One Wire (Dallas)
  - PS2Keyboard: driver per tastiere PS/2
  - Servo e Servotimer I: driver per il controllo di servomotori
  - Simple Message System: sistema di messaggistica tra PC e Arduino
  - SSerial2Mobile: per l'invio di SMS e email via cellulare
  - TLC5940: PWM controller a 12 bit per 16 canali
  - X10: driver per l'invio di segnali X10 su linea elettrica

setup( )

- setup() è la funzione eseguita **una sola volta** all'avvio del programma
- È responsabile delle impostazioni iniziali:
  - gestione dei PIN di ingresso/uscita
  - configurazione della porta seriale
  - inizializzazione librerie

```
void setup()
{
   // PIN 13 di uscita
   pinMode(13, OUTPUT);

    // PIN 2 di ingresso
    pinMode(2, INPUT)
}
```

LAB Open Media Center
SARDEGNA DISTRICT

loop()

- È il "cuore" di uno sketch: è la funzione eseguita ciclicamente finché Arduino è alimentato

```
void loop()
{
    readSensorData();

    if (temp > 100)
    {
        powerOnFan();
    }
    else if (temp < 40)
    {
        powerOffFan();
    }

    // riparte da capo...
}
```

Prima di cominciare...

- I circuiti di seguito descritti sono alimentati a bassa tensione. Questo riduce i rischi di danni alle persone e agli oggetti, ma è necessario adottare opportune cautele:
  - i componenti elettronici hanno estremità metalliche accuminate che possono lacerare la cute se premuti con forza o provocare abrasioni se proiettati verso gli occhi
  - Arduino dispone di un circuito di protezione della porta USB: è comunque consigliato, prima di effettuare modifiche al prototipo, scollegare la scheda dal computer
  - se si utilizza un saldatore, attenersi scrupolosamente al manuale d'uso, avendo cura di tenere lontana la punta calda dai cavi di alimentazione

LAB Open Media Center
SARDEGNA DISTRICT

Serie e parallelo

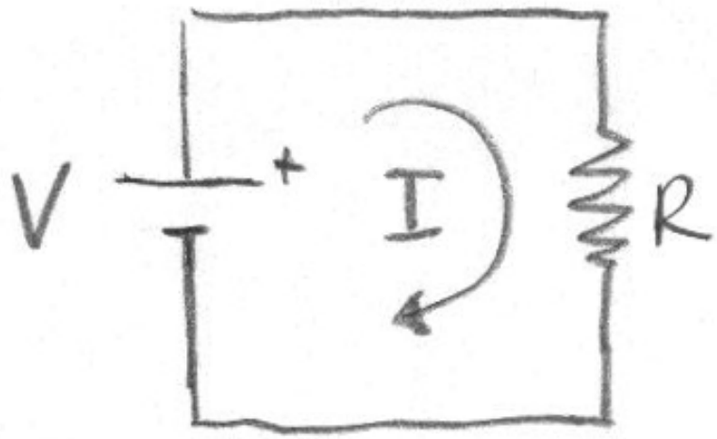
- Due componenti bipolari (con due soli terminali) si dicono connessi in **serie** quando l'uscita del primo è connessa all'ingresso del secondo, ovvero quando sono attraversati dalla stessa corrente. Due componenti in serie hanno un solo terminale in comune.

- Due componenti bipolari si dicono connessi in **parallelo** quando sono soggetti alla stessa tensione e hanno in comune ingressi e uscite.

LAB Open Media Center
SARDEGNA DISTRICT

Resistore

- Componente passivo che provoca una caduta di tensione
- La corrente circolante nel circuito è proporzionale alla tensione e inversamente proporzionale alla resistenza (misurata in Ohm)



$$I = \frac{V}{R}$$

Codici dei colori

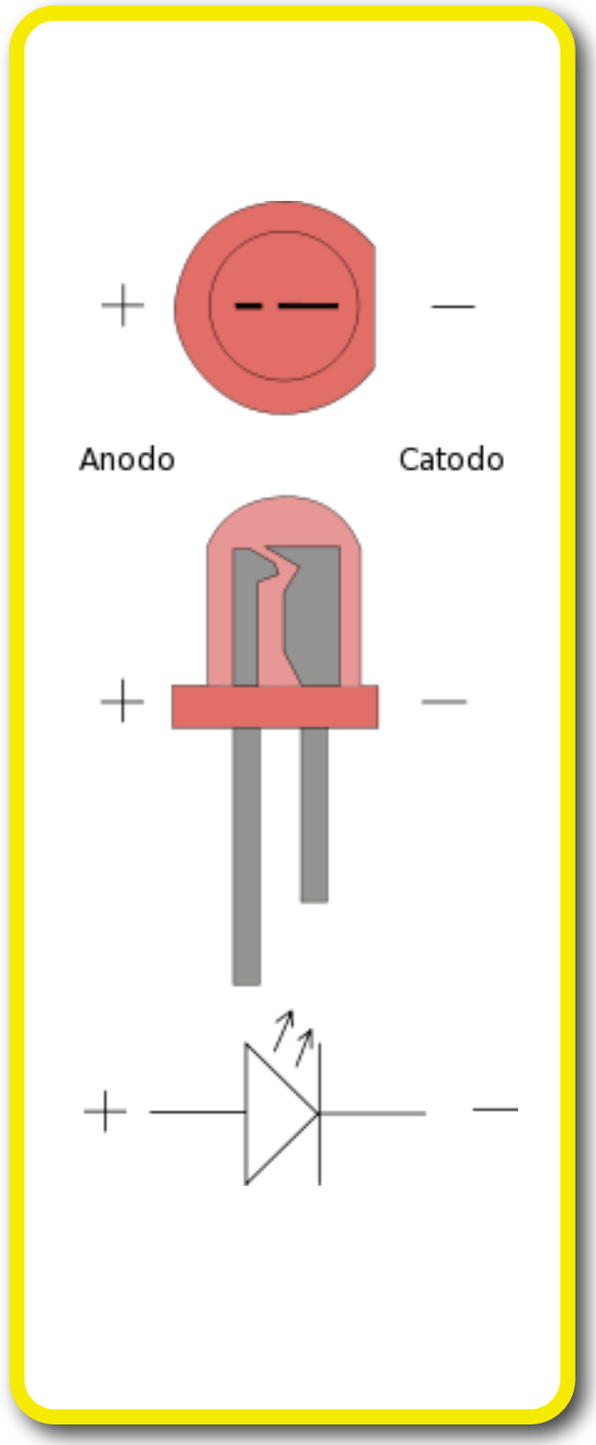
- Il valore della resistenza di un resistore è inciso sul corpo del componente attraverso una sequenza di bande colorate
- Ad ogni colore è assegnato una cifra, mentre ad ogni banda

Colore	1° Anello	2° Anello	3°Anello	4° Anello
	Cifra 1	Cifra2	Moltiplicatore	Tolleranze
-	-	-	-	± 20%
argento	-	-	10^{-2}	± 10%
oro	-	-	10^{-1}	± 5%
nero	0	0	10^{0}	-
marrone	1	1	10^{1}	± 1%
rosso	2	2	10^{2}	± 2%
arancio	3	3	10^{3}	-
giallo	4	4	10^{4}	-
verde	5	5	10^{5}	± 0,5%
blu	6	6	10^{6}	± 0,25%
viola	7	7	10^{7}	± 0,1%
grigio	8	8	10^{8}	± 0,05%
bianco	9	9	10^{9}	-

LAB Open MediaCenter
SARDEGNA DISTRICT

LED

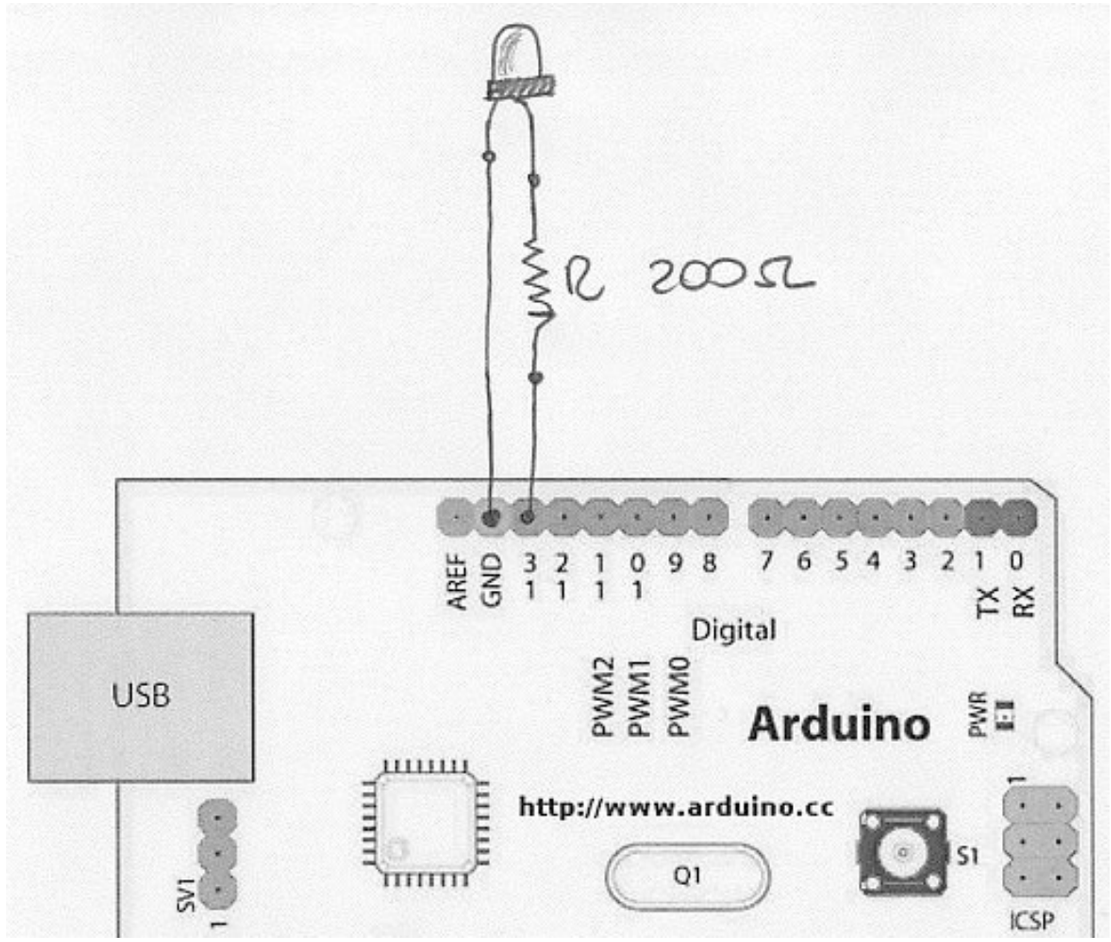
- Un **LED** (light-emitting diode) è un dispositivo a semiconduttore che emette **luce** quando attraversato da una corrente continua



- Una corrente eccessiva può **danneggiarlo**: è consigliato alimentarlo attraverso una resistenza che limiti la corrente circolante a 20mA

Hello World

- Per limitare la corrente che attraversa il LED, è necessario porre una resistenza in serie
- Essendo le uscite a 5V, è sufficiente porre una resistenza di 220 ohm



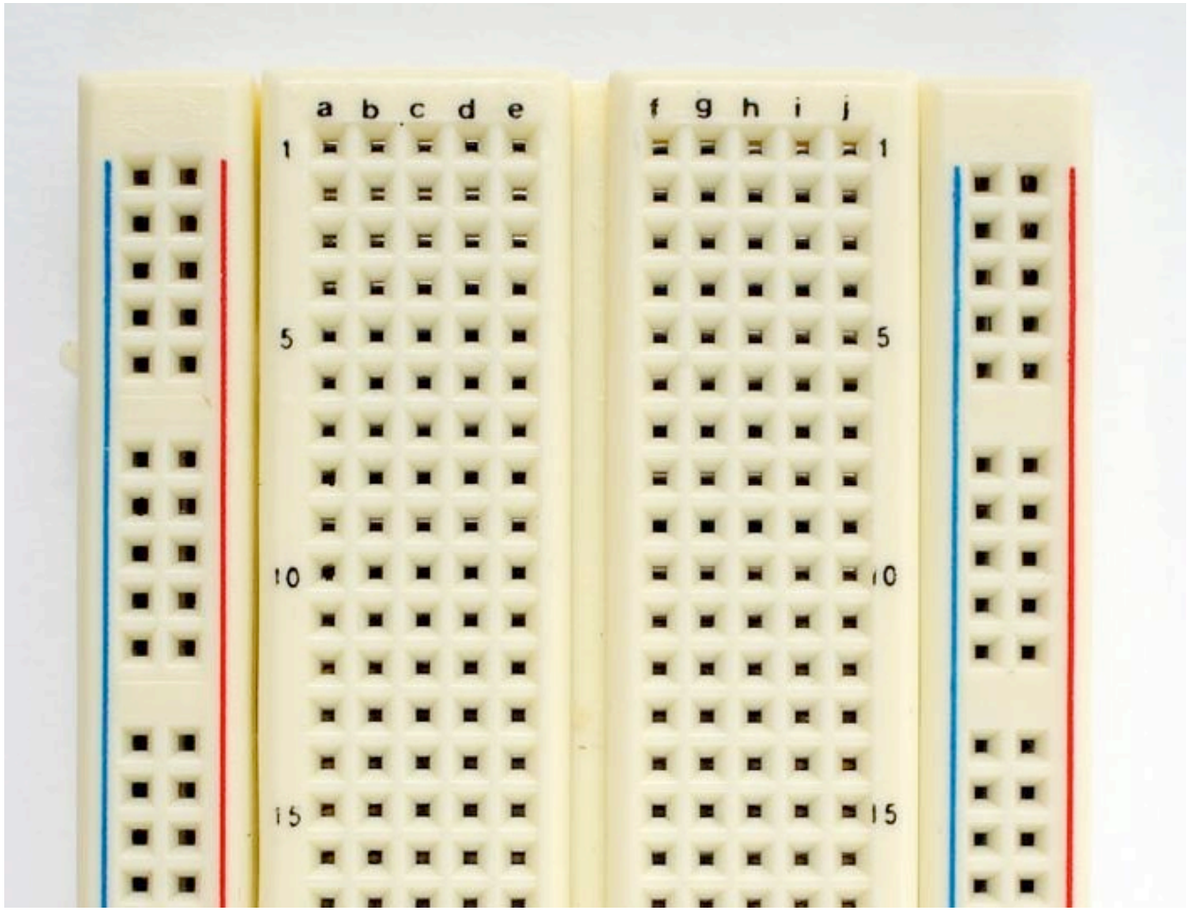
Cablaggio dei circuiti

- La realizzazione dei circuiti richiede il collegamento elettrico tra i componenti ed un adeguato **isolamento** delle connessioni dal resto dell'ambiente
- Nei casi più semplici è possibile usare "collegamenti volanti" con morsettiere o piccole saldature
- Al crescere della complessità del circuito e volendolo modificare spesso con la certezza di riciclare i componenti, è consigliabile utilizzare schede sperimentali (**breadboard**)

LAB Open MediaCenter
SARDEGNA DISTRICT

Breadboard

- Consente la realizzazione di circuiti elettrici senza saldature
- Il cablaggio avviene collocando i componenti ed effettuando i collegamenti con piccoli spezzoni di filo metallico

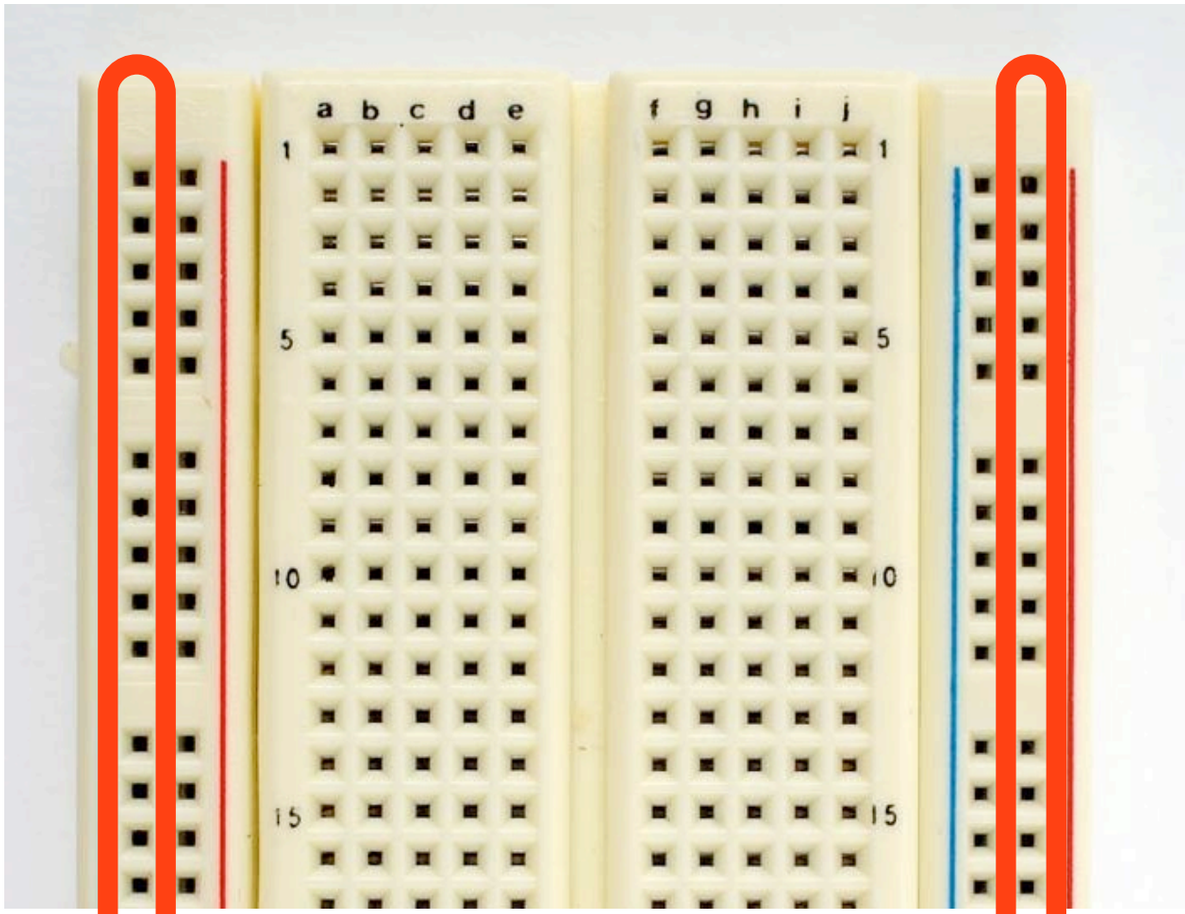


LAB Open Media Center
SARDEGNA DISTRICT

Breadboard

Linee di alimentazione

- Le linee di **alimentazione** sono continue verticalmente e consentono di distribuire massa e tensione positiva lungo tutto il circuito

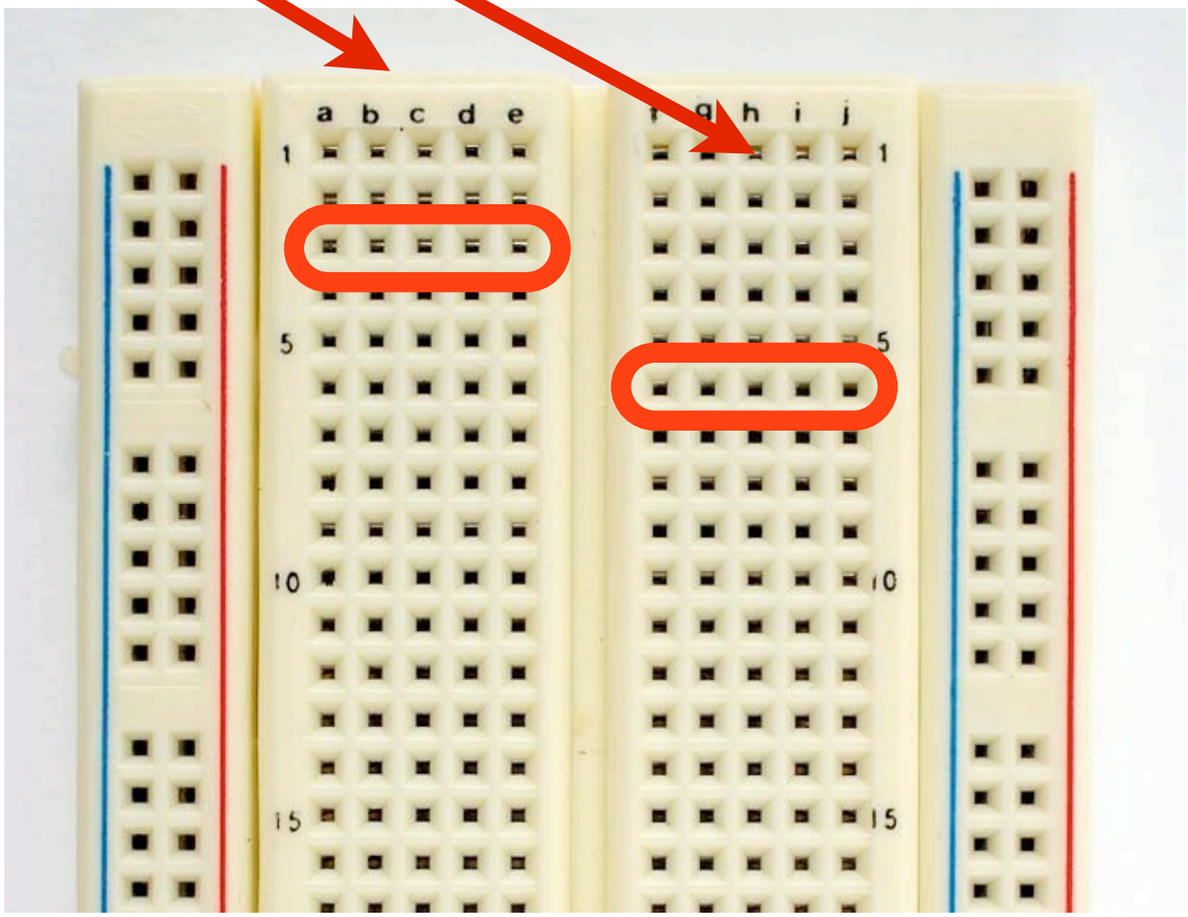


LAB Open Media Center
SARDEGNA DISTRICT

readboard

Bus dei componenti

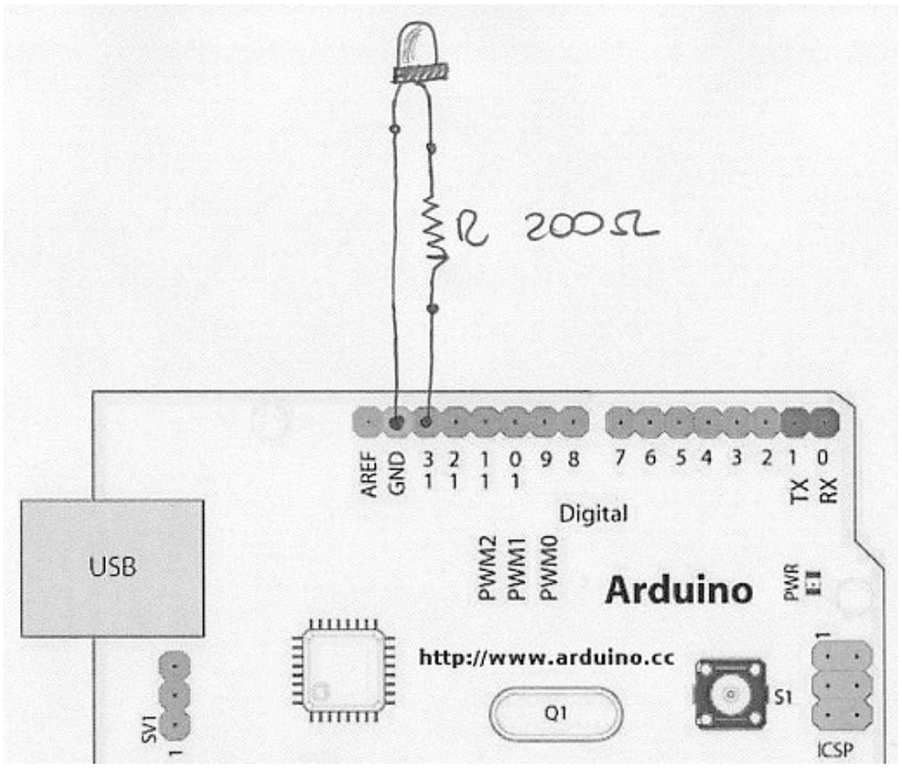
- Il **bus** dei componenti si estende sulle righe orizzontali
- Un componente può essere collocato “a cavallo” tra due sezioni per sfruttare due file di pin paralleli

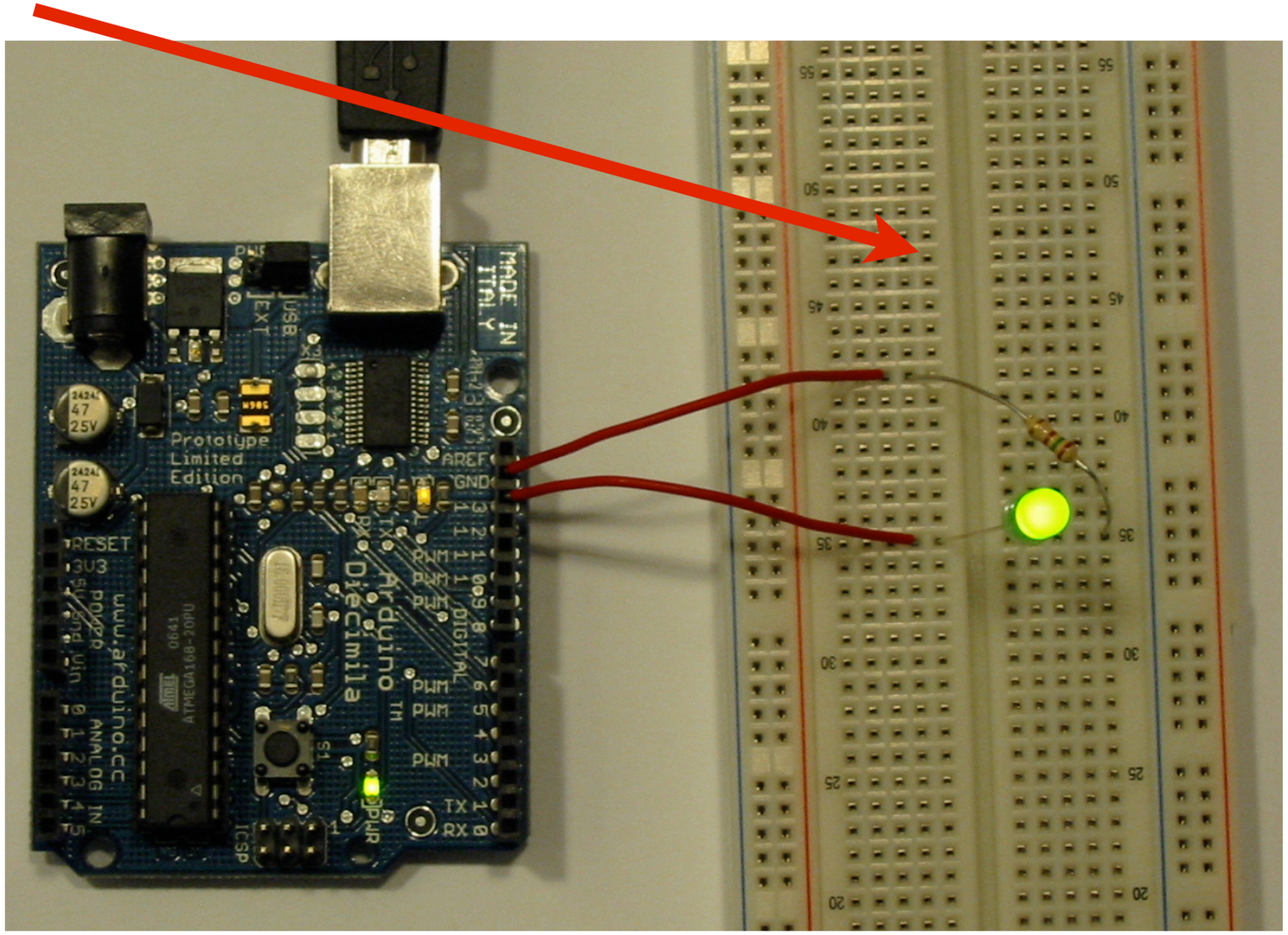


LAB Open Media Center
SARDEGNA DISTRICT

Hello Arduino

Nota: la resistenza è
stata posta “a valle” del LED.

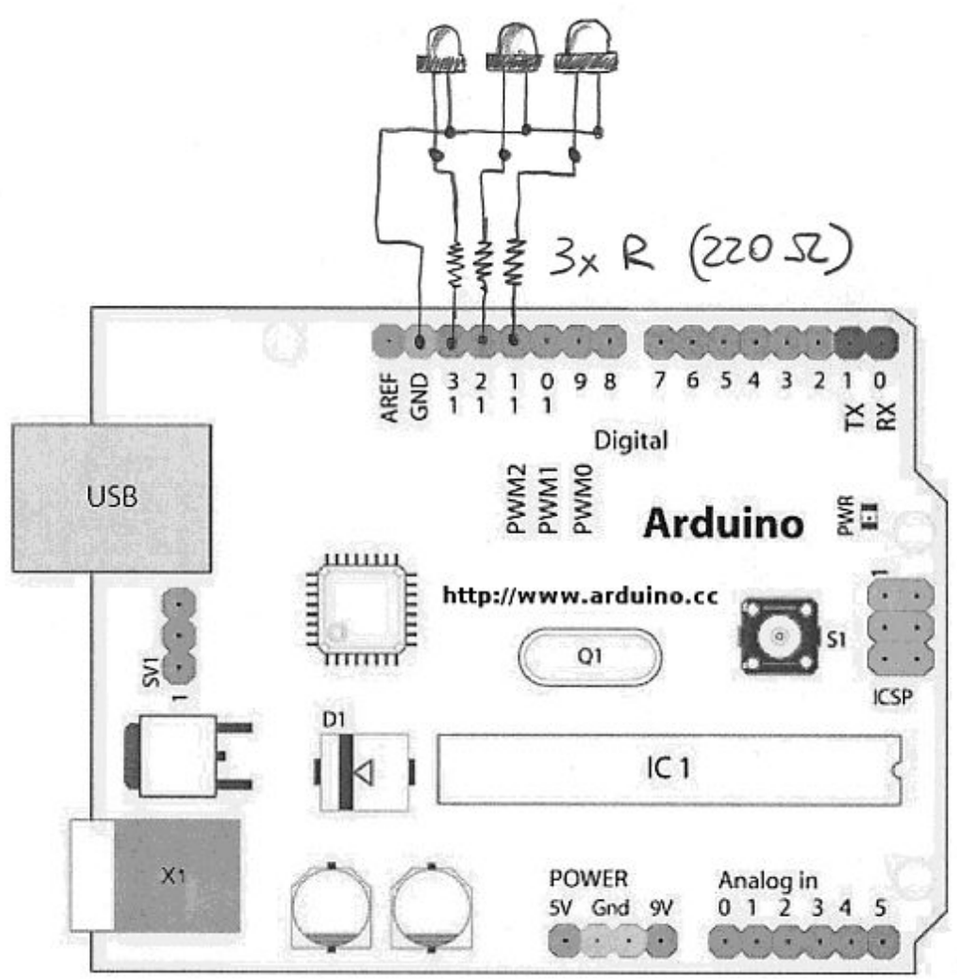




LAB Open Media Center
SARDEGNA DISTRICT

Un semaforo

- Tre LED sono accesi in sequenza:
  - verde acceso, gli altri spenti
  - dopo 2 secondi, si accende il giallo
  - dopo due secondi, si spengono verde e giallo, si accende il rotto
  - dopo due secondi, riparti da capo



Sketch [1]

- Il primo passo consiste nel definire tre variabili corrispondenti ai PIN che si vogliono utilizzare e nell'impostarli come uscite digitali

```
int GREEN = 13;
int YELLOW = 12;
int RED = 11;

void setup()
{
  pinMode(GREEN, OUTPUT);
  pinMode(YELLOW, OUTPUT);
  pinMode(RED, OUTPUT);
}
```

LAB Open Media Center
SARDEGNA DISTRICT

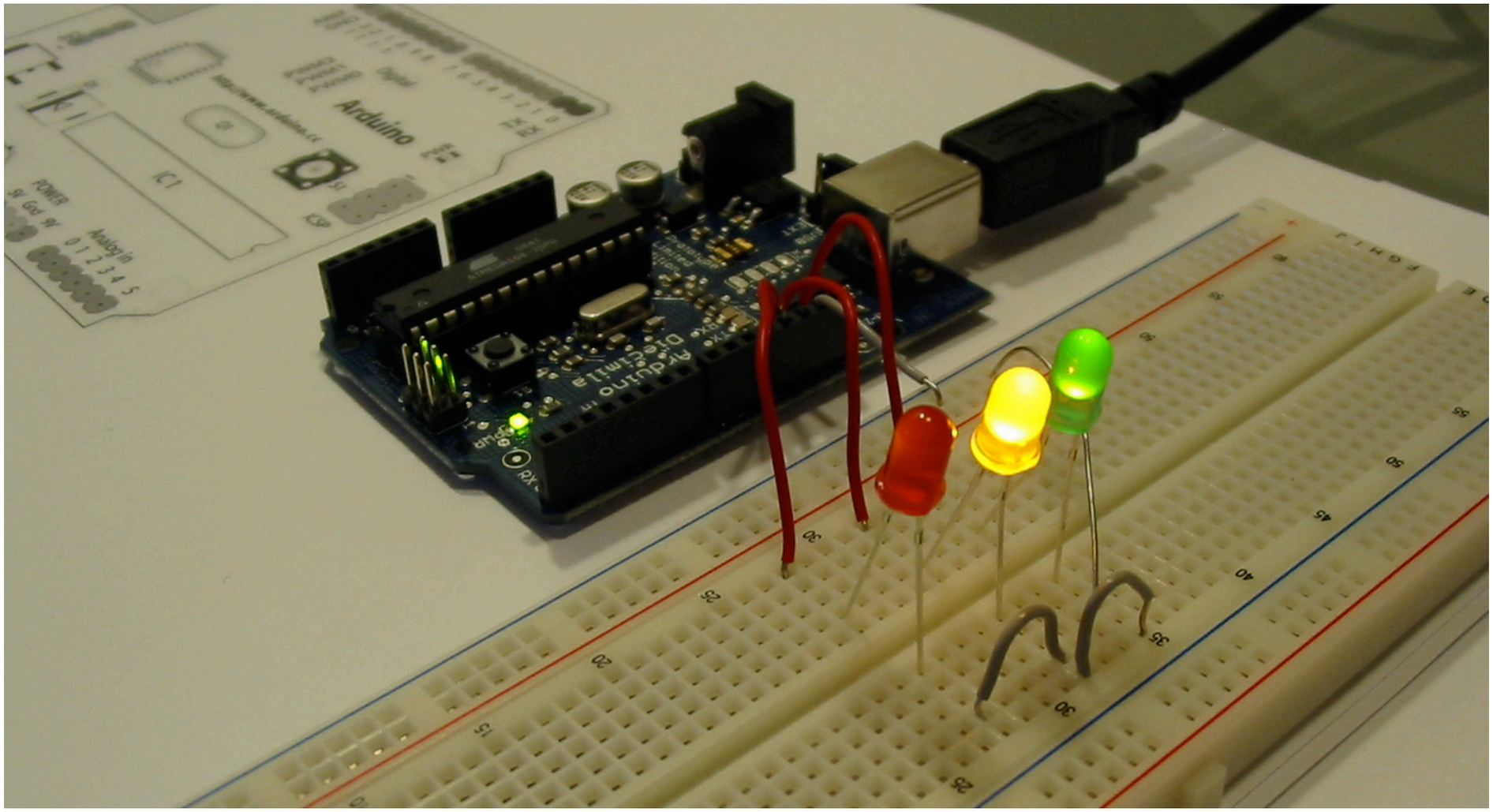
Sketch [2]

- Nel loop() avviene:
  - si spengono rosso e giallo, si accende il verde; si aspetta due secondi
  - si accende il giallo; si aspetta altri due secondi
  - si accende il rosso, si spengono gli altri; si aspetta due secondi

```
void loop()
{
  digitalWrite(RED, LOW);
  digitalWrite(YELLOW, LOW);
  digitalWrite(GREEN, HIGH);
  delay(2000);

  digitalWrite(YELLOW, HIGH);
  delay(2000);

  digitalWrite(RED, HIGH);
  digitalWrite(YELLOW, LOW);
  digitalWrite(GREEN, LOW);
  delay(2000);
}
```

LAB Open MediaCenter
SARDEGNA DISTRICT

Semaforo



Input/Output

I/O analogico e digitale, porte seriali

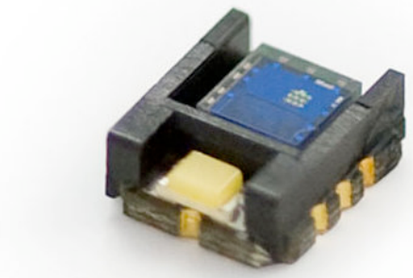
LAB Open MediaCenter

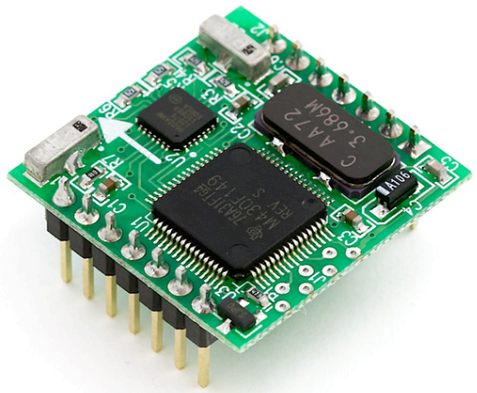
SARDEGNA DISTRICT

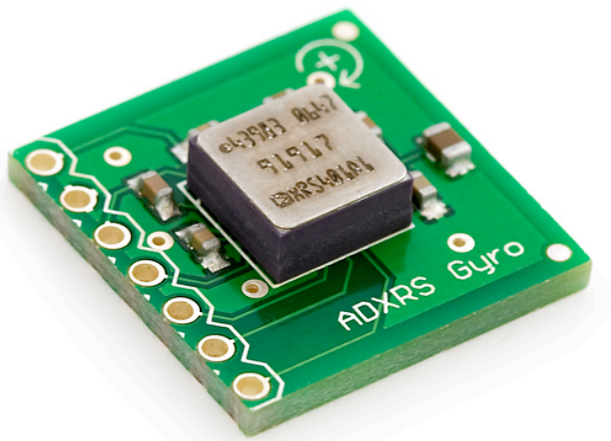
CRS4

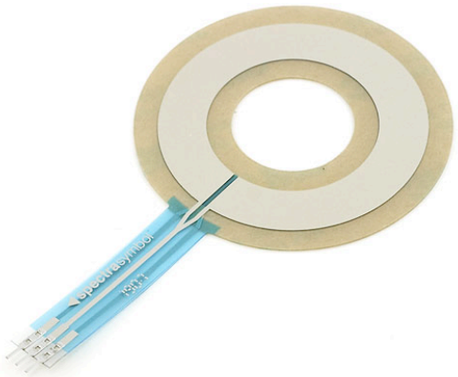
LAB Open MediaCenter
SARDEGNA DISTRICT

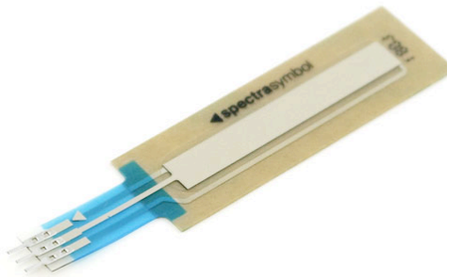
Sensori

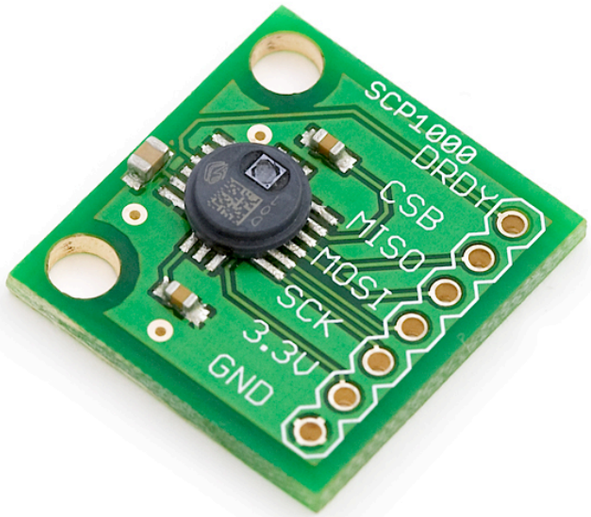


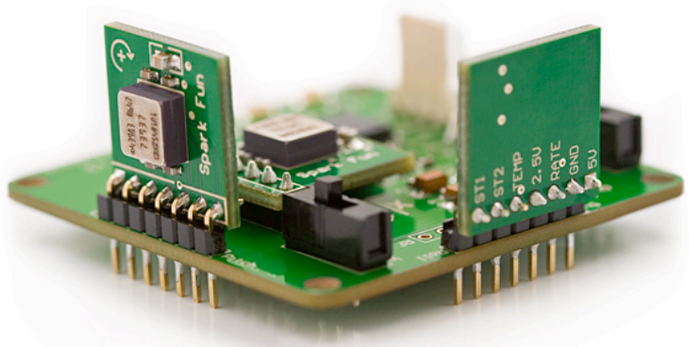












LAB Open Media Center
SARDEGNA DISTRICT

Analogico vs Digitale

- Un segnale **analogico** può assumere qualsiasi valore (all'interno di un range noto). Con notevole semplificazione, si può pensare che siano "analogiche" tutte le grandezze fisiche misurabili nell'ambiente

- Un segnale **digitale** può assumere due soli stati (HIGH - LOW), corrispondenti a due livelli di tensione convenzionali (ad esempio, 0-5V). Con simile semplificazione, si può pensare che siano "digitali" tutte le informazioni scambiate tra componenti logici (microprocessori, memorie, interfacce di rete, display...)

LAB Open Media Center
SARDEGNA DISTRICT

Analogico vs Digitale

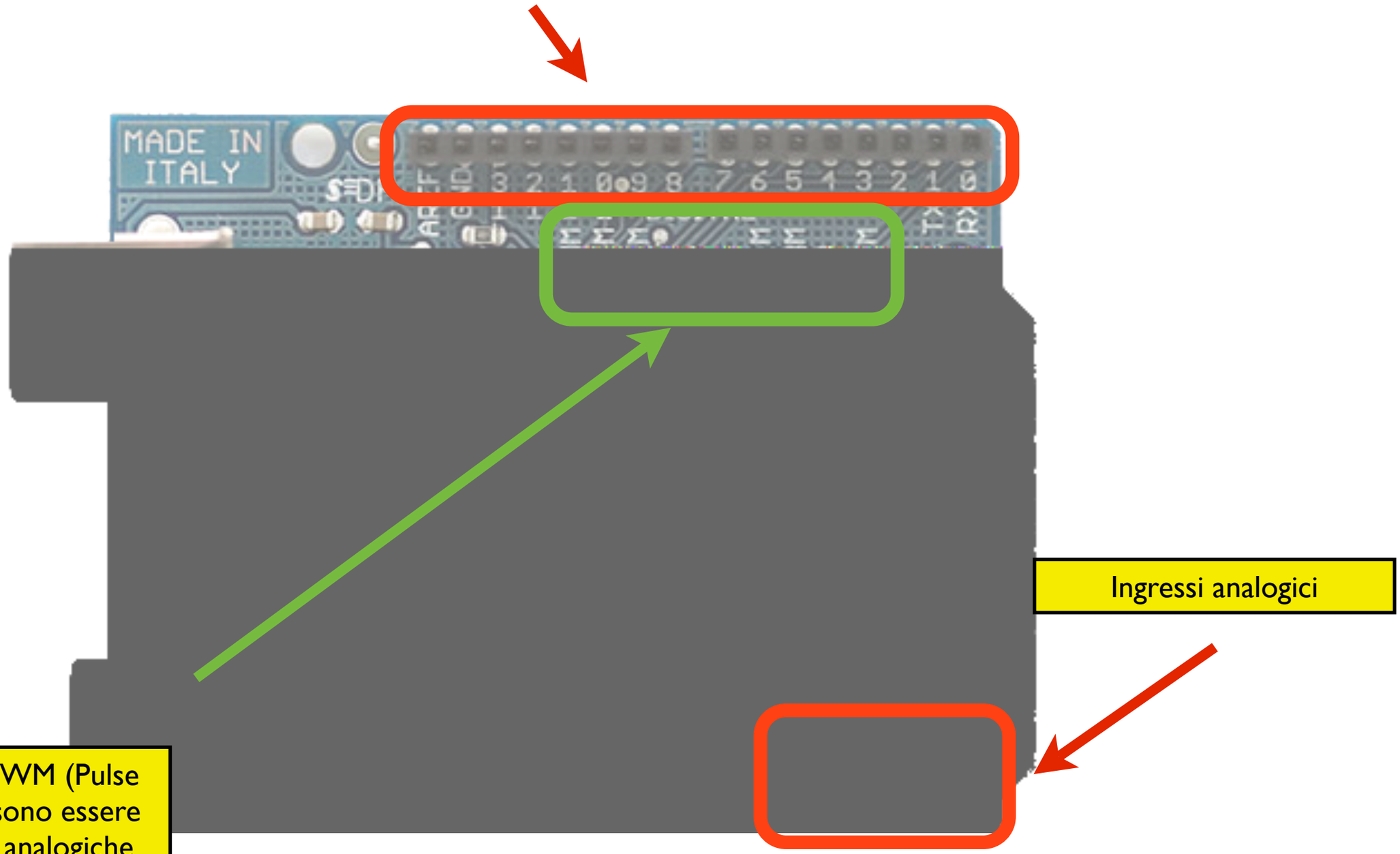
- Affinché possa essere letto ed elaborato da Arduino, il segnale analogico deve essere **campionato**, ovvero **convertito** in in una sequenza di bit che ne esprime l'ampiezza

- Un segnale digitale è immediatamente "leggibile" non appena ne è stato discriminato il livello (HIGH - LOW).

LAB Open MediaCenter
SARDEGNA DISTRICT

Interfacce

Input/Output digitali



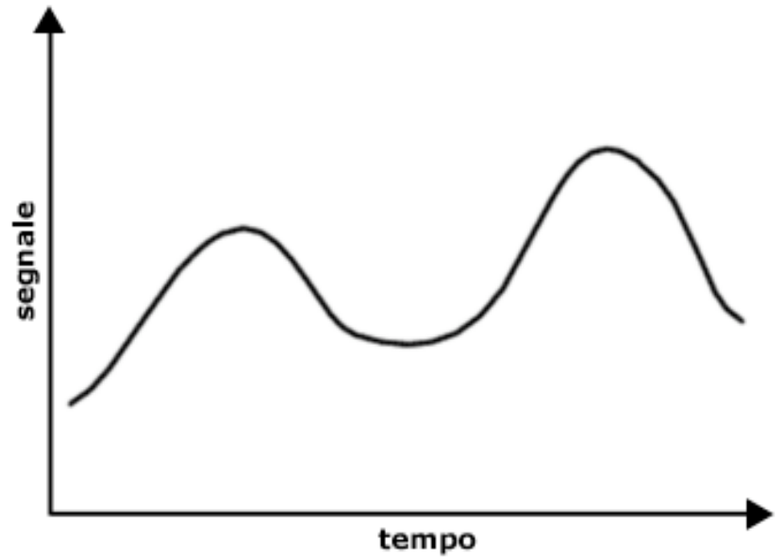
Ingressi analogici

I 6 PIN digitali marcati PWM (Pulse Width Modulation) possono essere configurati come uscite analogiche

Analogico vs Digitale

- Del segnale analogico interessa leggere il valore istantaneo, opportunamente campionato

- Del segnale digitale occorre sapere solo lo stato alto o basso



Analogico vs Digitale

- I PIN digitali devono essere impostati per input o output
- Lettura e scrittura avvengono attraverso funzioni dedicate:
  - digitalWrite() e digitalRead()
  - analogWrite() e analogRead()

```
pinMode(13, OUTPUT);

pinMode(2, INPUT);


digitalWrite(13, HIGH)

int button = 0;
button = digitalRead(2);

int temp;
temp = analogRead(0);
```

LAB Open Media Center
SARDEGNA DISTRICT

Input

- Arduino dispone di linee di ingresso analogiche e digitali per l'interfacciamento a sensori
- Esempi di componenti di input:
  - pulsante (apertura/chiusura di un circuito)
  - potenziometro (variazione di resistenza in funzione di rotazione/traslazione)
  - fotoresistenza: (resistenza in funzione dell'intensità luminosa)
  - sensore di temperatura
  - accelerometro
  - lettore RFID

LAB Open Media Center
SARDEGNA DISTRICT

Output

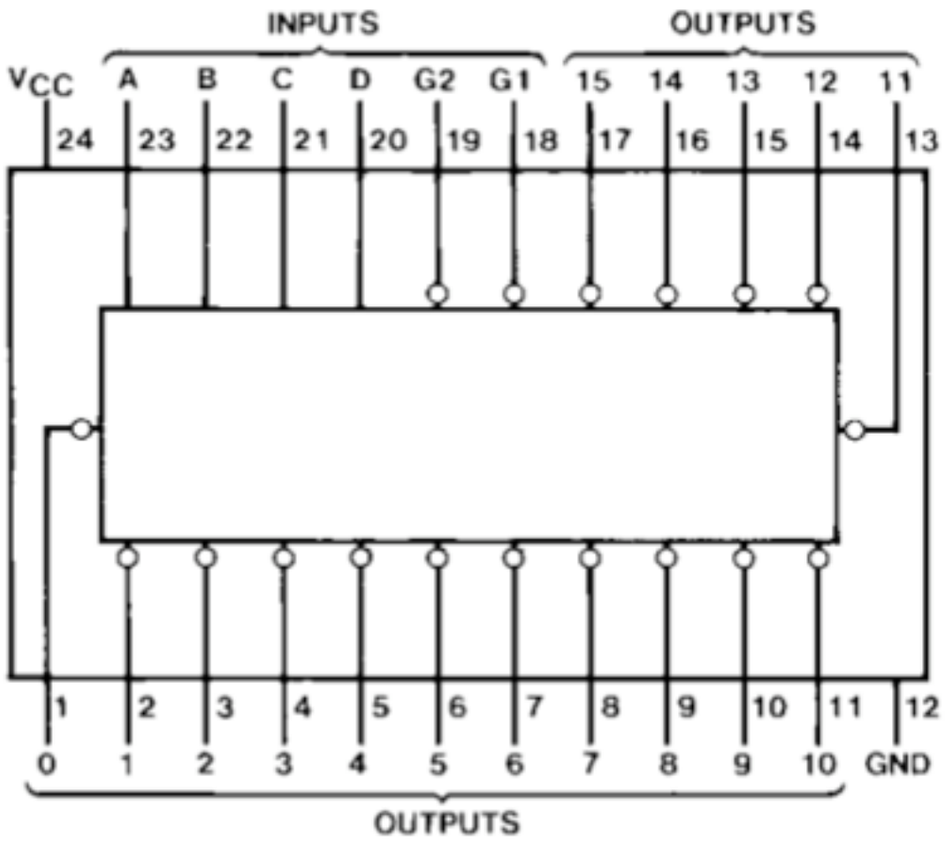
- Arduino dispone di linee di uscita digitali (livello alto/basso) e analogiche (PWM), per il controllo di attuatori
- Esempi di componenti di output:
  - LED (luce)
  - display LCD (testo e grafica)
  - buzzer (suono)
  - motori (direct/servo/stepper, movimento rotatorio/traslatorio)
  - qualsiasi dispositivo controllabile con un driver dedicato

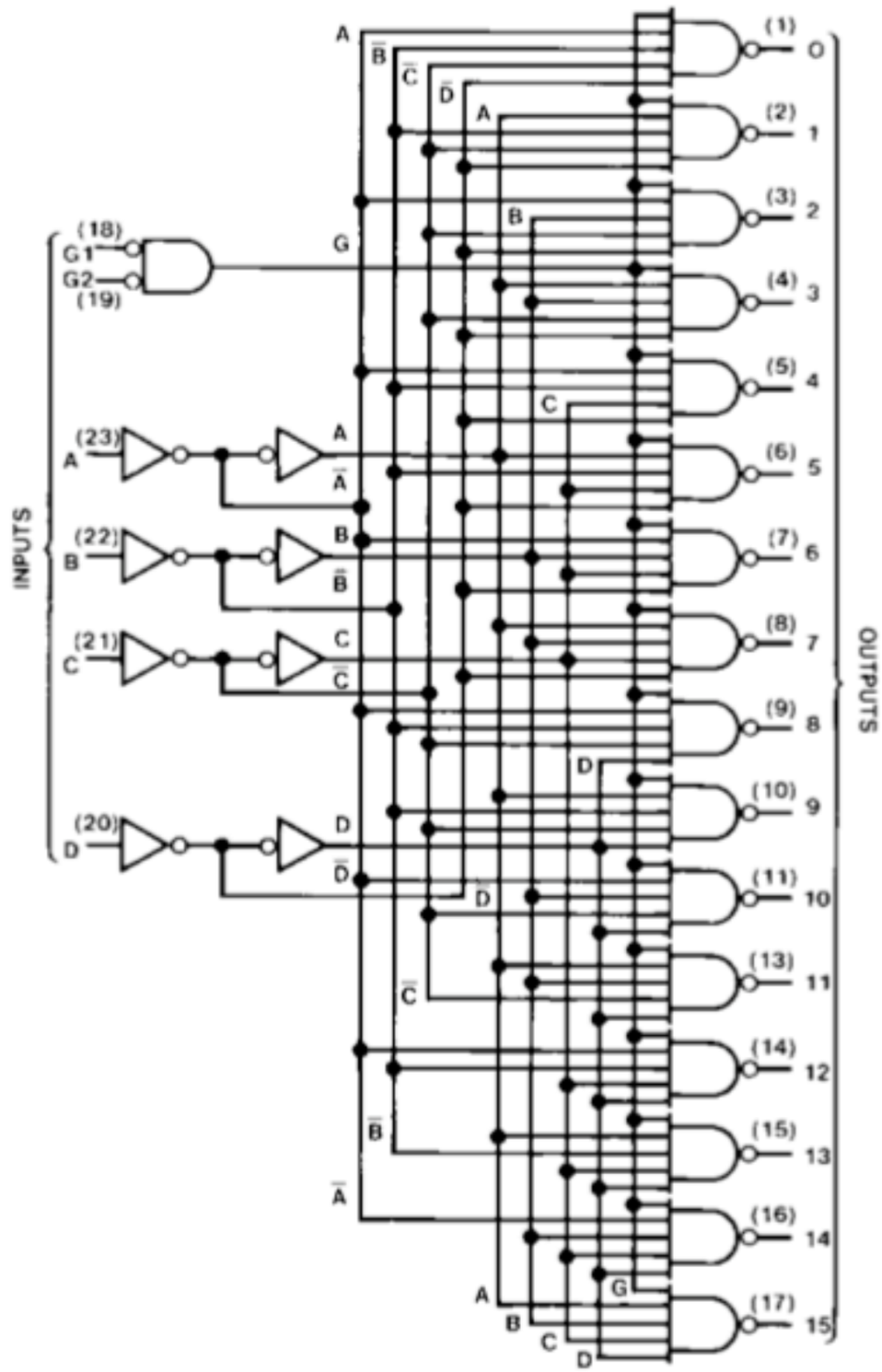
Multiplexer/Demultiplexer

- Per progetti di media complessità i PIN di I/O possono essere insufficienti
- Attraverso opportuni multiplexer/demultiplexer analogici e digitali è possibile "moltiplicare" il numero di ingressi ed uscite di Arduino:
  - N PIN di controllo abilitano 2^N linee fisiche
  - 1 PIN dati è commutato sulle 2^N linee fisiche
- Utile per:
  - controllare matrici di LED
  - leggere in rapida sequenza lo stato di numerosi switch

LAB Open MediaCenter
SARDEGNA DISTRICT

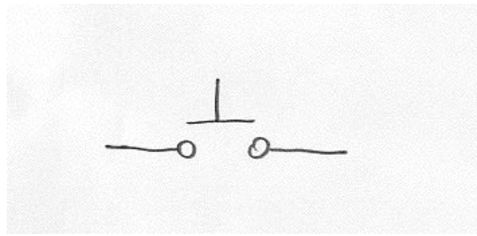
Multiplexer/Demultiplexer

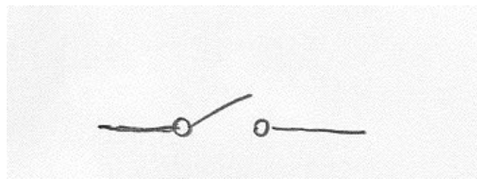


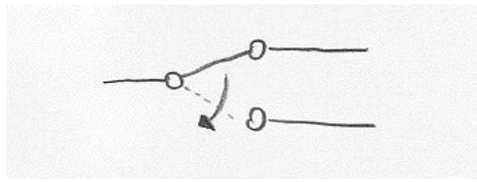


Interruttori

- Pulsante
  - chiude il circuito finché premuto, per poi riaprirlo al rilascio
- Interruttore
  - chiude o apre il circuito in maniera permanente
- Commutatore
  - devia la corrente tra due rami; l'azione è permanente, fino alla successiva commu



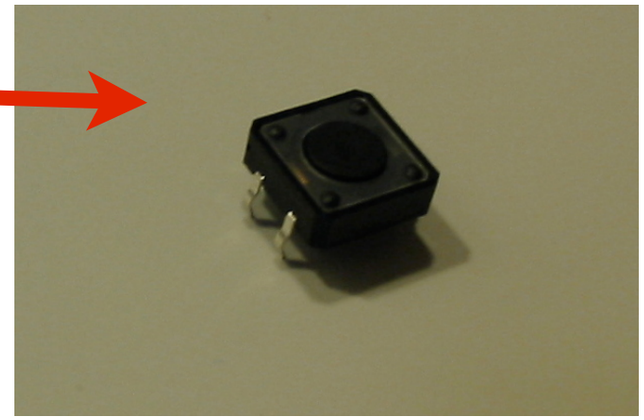


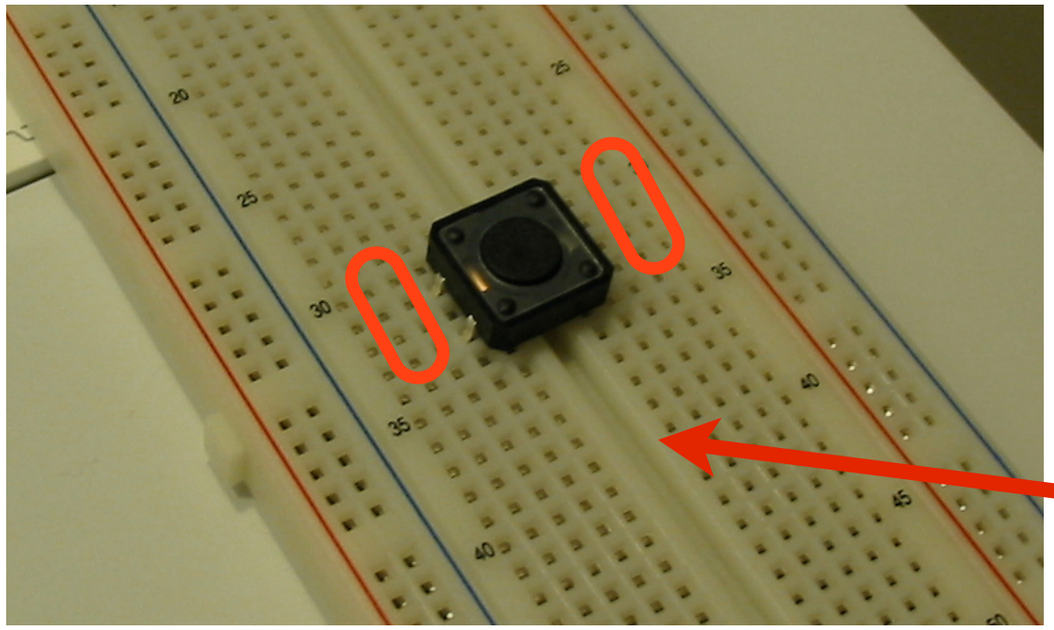


LAB Open MediaCenter
SARDEGNA DISTRICT

Pulsante

Il pulsante chiude il circuito ai capi delle coppie di PIN più vicine

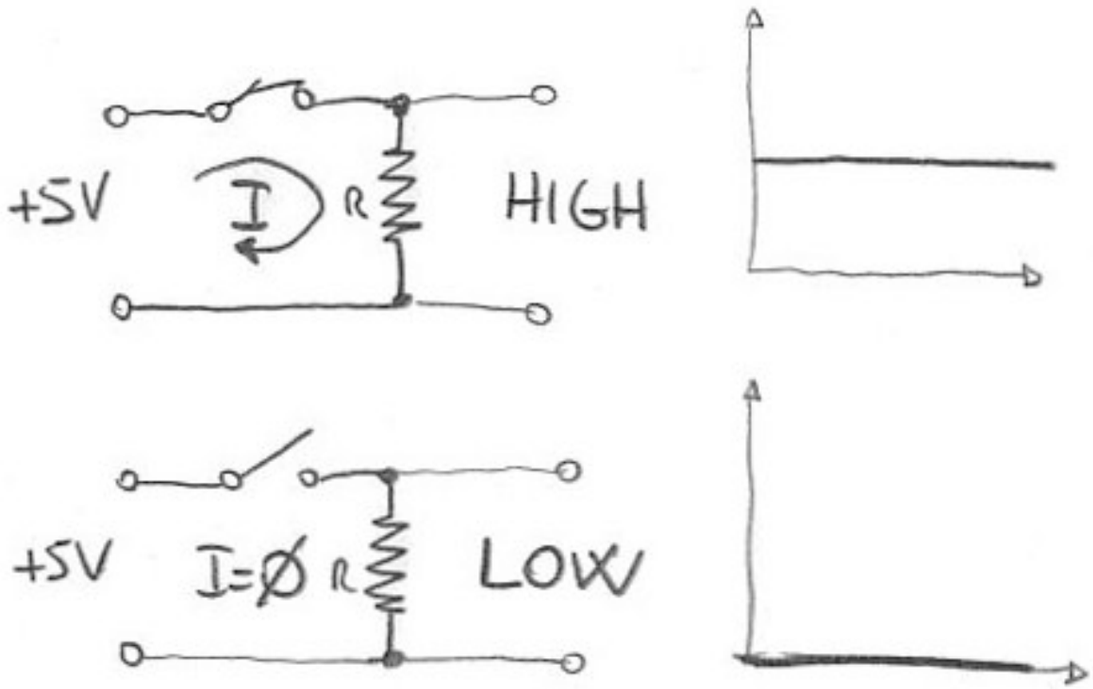




Il pulsante si pone a cavallo dei due bus. Alla pressione del pulsante, su entrambi i bus sono cortocicuitati i PIN

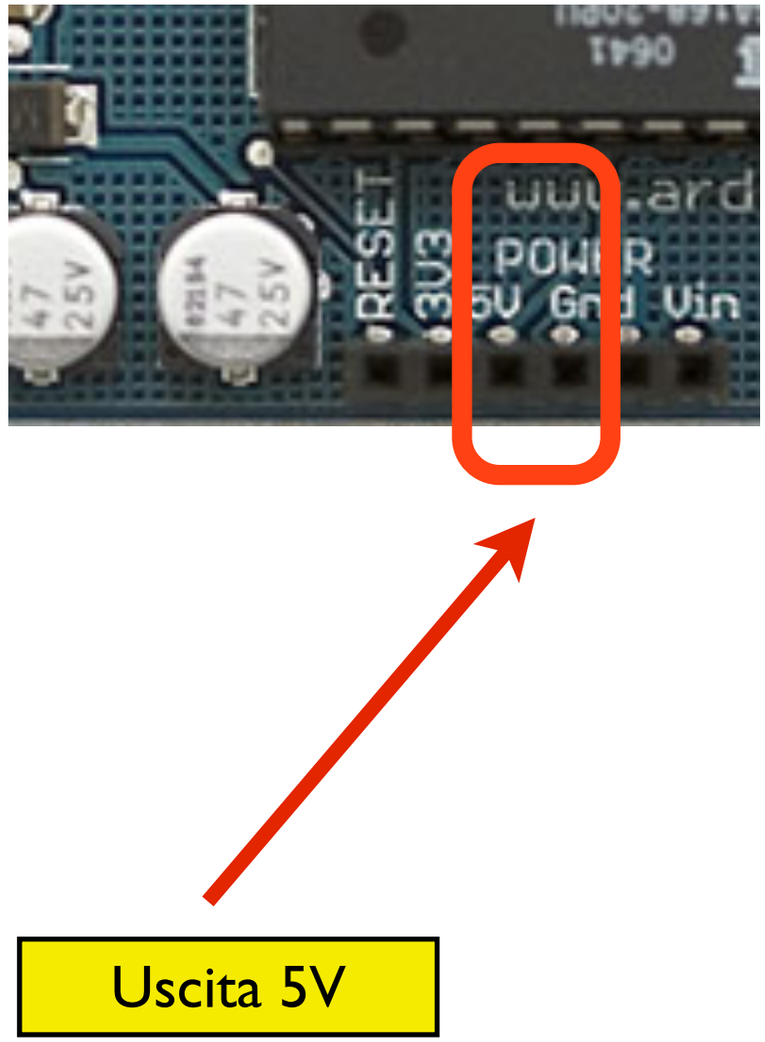
Inviare i livelli logici

- HIGH: il circuito è chiuso, la corrente circola all'interno della resistenza ai cui capi è presente la differenza di potenziale corrispondente al livello alto
- LOW: il circuito è aperto, non vi è alcune ddp ai capi della resistenza



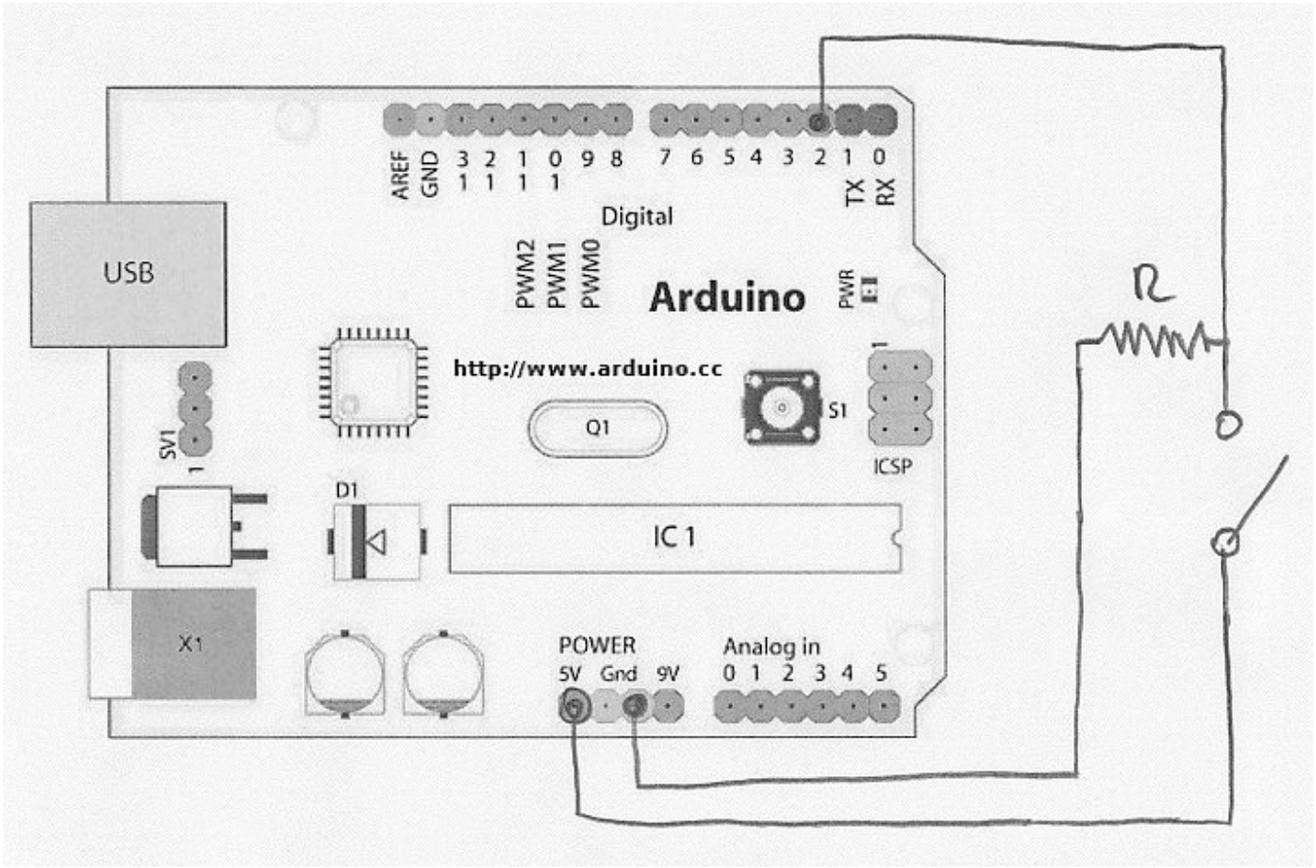
Alimentare il circuito

- Arduino dispone di due PIN a 5V e 3.3V che possono essere utilizzati per alimentare piccoli componenti del circuito realizzato sulla breadboard



LAB Open MediaCenter
SARDEGNA DISTRICT

Il circuito



Input digitale

- Un ingresso digitale può assumere i due valori LOW (basso) o HIGH (alto)

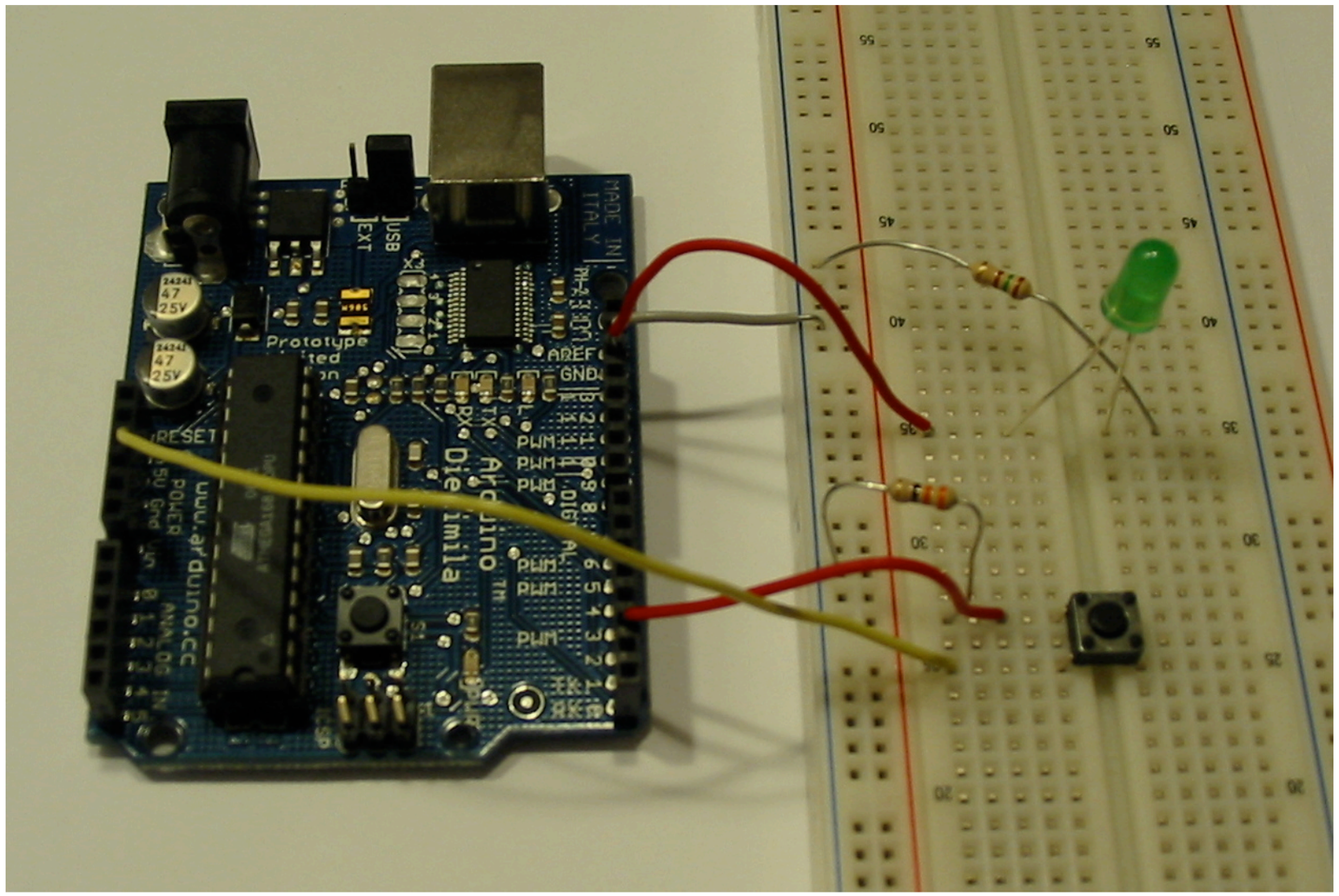
- I due valori corrispondono rispettivamente ad una tensione nulla (GND) o positiva (5V nominali)

```
int BTN_PIN = 2;
int LED_PIN = 13;
int state = LOW;

void setup()
{
    pinMode(BTN_PIN, INPUT);
    pinMode(LED_PIN, OUTPUT);
}

void loop()
{
    int state =
digitalRead(BTN_PIN);
    digitalWrite(LED_PIN,
state);
}
```

LAB Open MediaCenter

SARDEGNA DISTRICT

Il prototipo



Variante (I)

- Il LED lampeggia con un intervallo di mezzo secondo

- Alla pressione del pulsante, il LED lampeggia più velocemente

```
int BTN_PIN = 2;
int LED_PIN = 13;
int state = LOW;
int ledDelay = 500;

void setup()
{
    pinMode(BTN_PIN, INPUT);
    pinMode(LED_PIN, OUTPUT);
}
```

Variante (II)

- Il LED lampeggia con un intervallo di mezzo secondo
- Alla pressione del pulsante, il LED lampeggia più velocemente

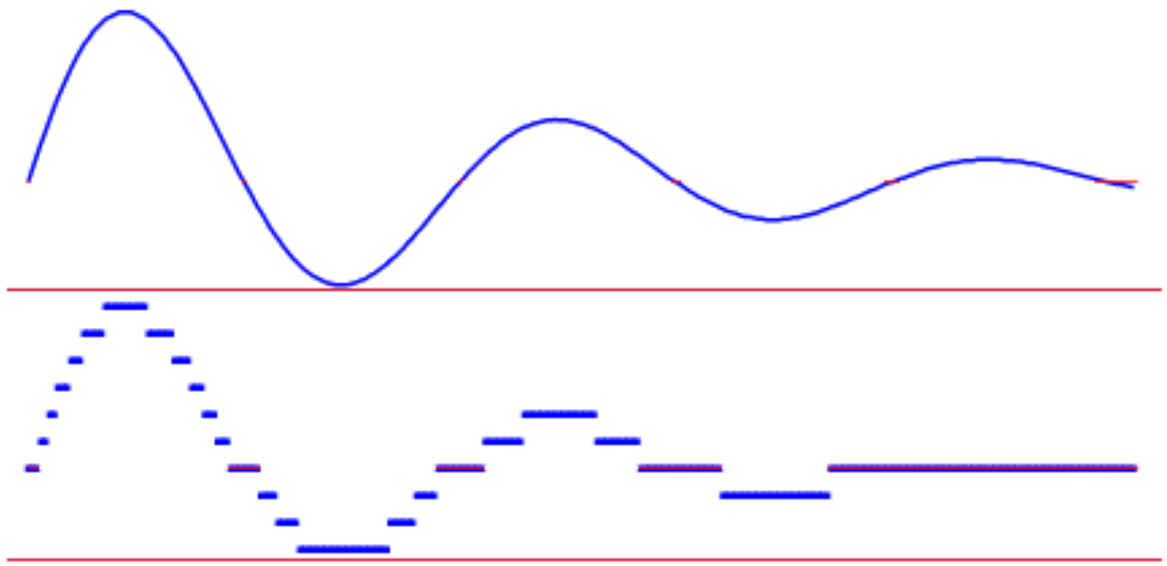
```
void loop()
{
    int state =
digitalRead(BTN_PIN);
    if (state == HIGH)
    {
        ledDelay = 200;
    } else
    {
        ledDelay = 500;
    }
    digitalWrite(LED_PIN,
HIGH);
    delay(ledDelay);
    digitalWrite(LED_PIN,
LOW);
    delay(ledDelay);
}
```

LAB Open Media Center
SARDEGNA DISTRICT

Input analogico

- Attraverso gli ingressi analogici è possibile leggere tensioni comprese tra 0 e 5V
- analogRead() riceve come parametro il PIN da leggere e restituisce un valore tra 0 e 1023 (ogni unità a 4.9 mV)

```
int temp;

temp = analogRead(0);
```



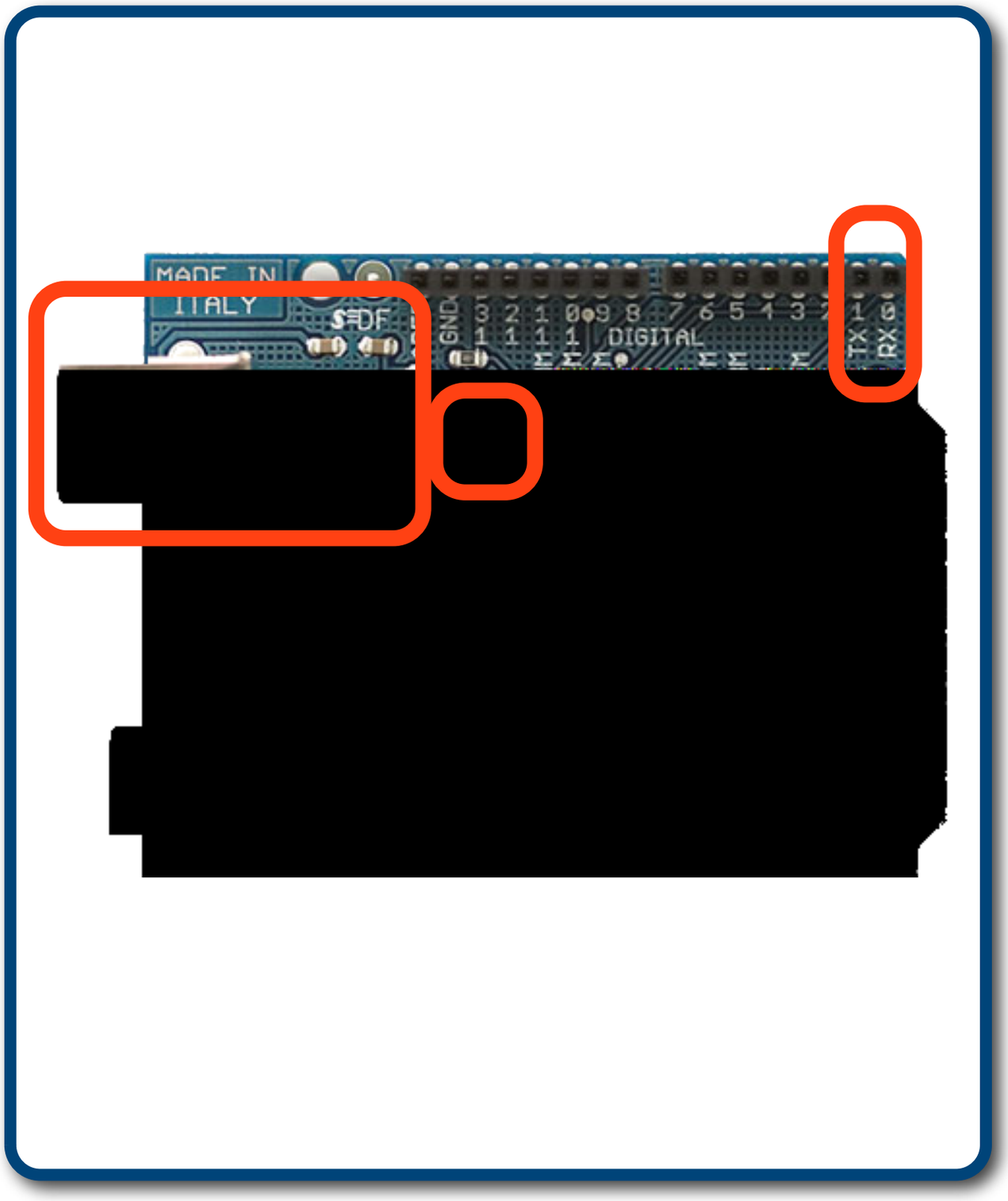
Cosa sta leggendo Arduino?

- La pressione di un pulsante (input digitale) può essere immediatamente verificata attraverso un output semplice: un LED acceso o spento
- Nel caso di un input analogico, il valore in ingresso appartiene ad un range ampio (1024 possibili valori) ed è dunque necessario predisporre un tipo di output diverso
- Sfruttiamo la porta seriale...

LAB Open Media Center
SARDEGNA DISTRICT

Serial port

- Il MCU Atmel ha una porta seriale TTL sui PIN 1 e 2
- Il transito di dati in ingresso e uscita è segnalato dai due LED TX ed RX
- La seriale è indirizzata anche sulla porta USB



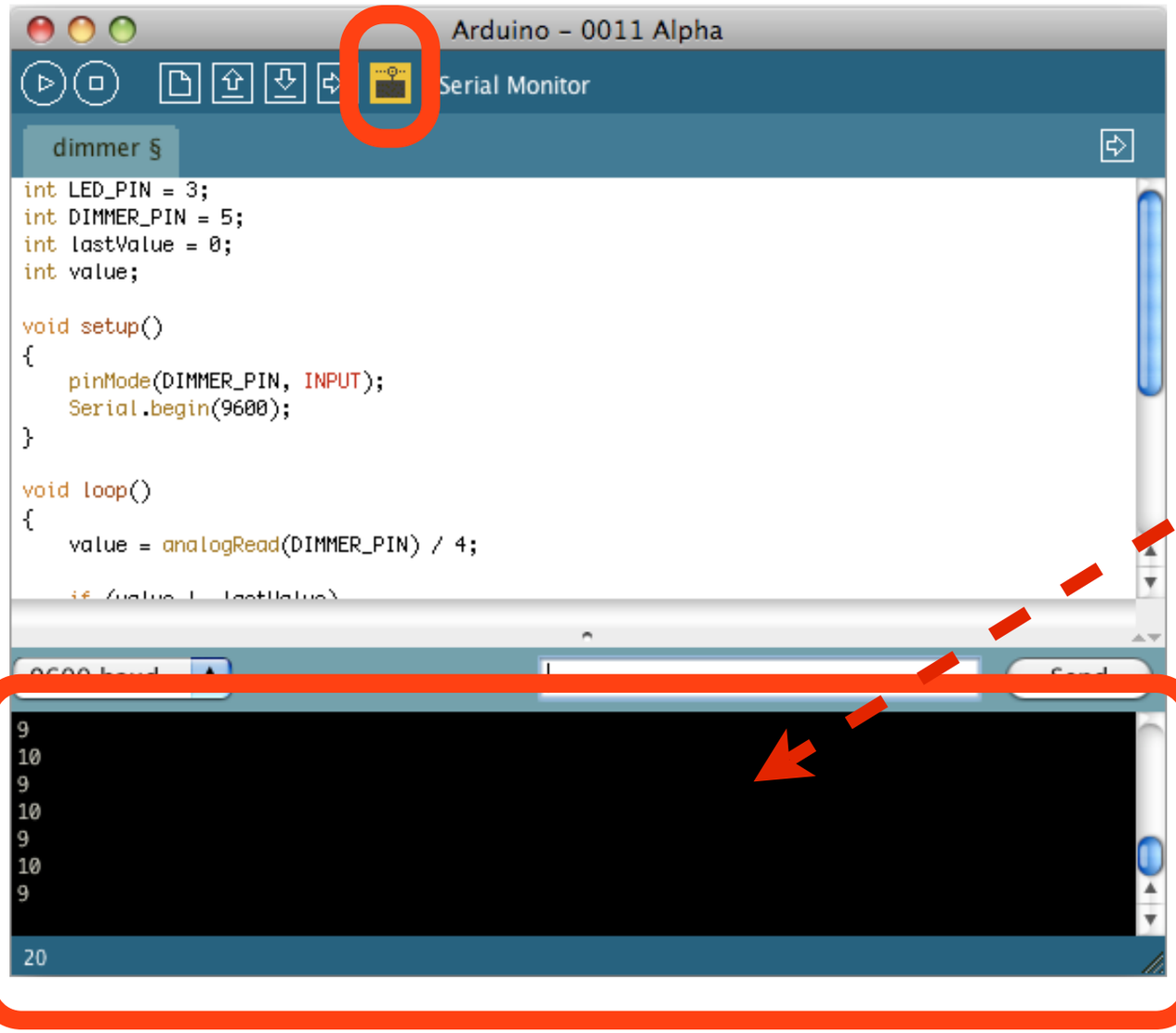
LAB Open Media Center
SARDEGNA DISTRICT

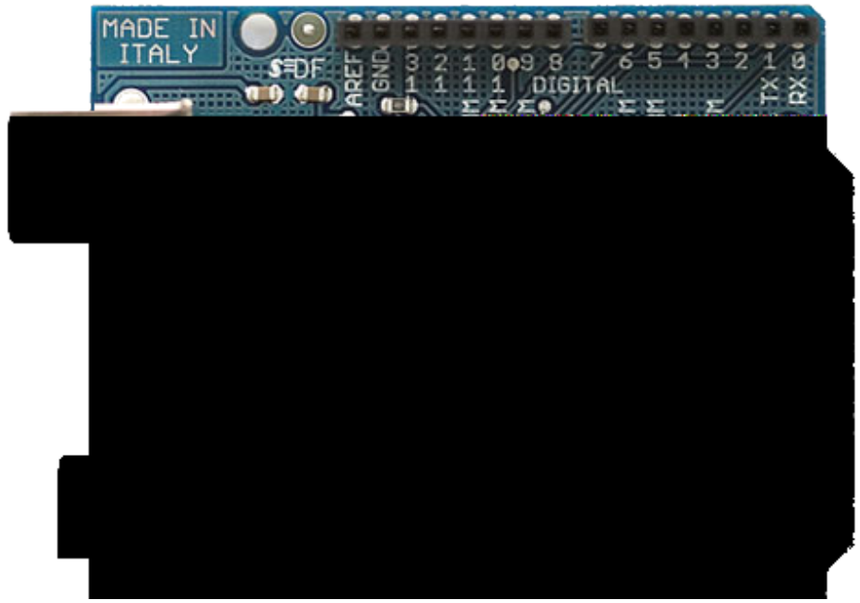
Serial port

- La connessione seriale (USB o Bluetooth) utilizzata per il caricamento degli sketch può essere utilizzara per la comunicazione con il PC **durante l'esecuzione** di un programma
- possibile dunque ricevere comandi (read) dal PC o inviare i dati provenienti dai sensori (write) al PC
- Attivando la **console seriale** nell'ambiente di sviluppo è possibile controllare in tempo reale l'esecuzione del programma inserendo opportune istruzioni in scrittura

LAB Open MediaCenter
SARDEGNA DISTRICT

Serial port





LAB Open Media Center
SARDEGNA DISTRICT

Serial port

- Serial.begin() apre la porta seriale specificando la velocità di comunicazione
- Serial.print() scrive un valore che sarà ricevuto dal PC
- Serial.read() legge un valore dal PC

```
// imposta la seriale
// alla velocità
// SPEED
Serial.begin(9600);


// stampa il valore
// sulla seriale
Serial.print(VALORE);


// legge un valore
// dalla seriale
int value = 0;
value = Serial.read();
```

LAB Open Media Center
SARDEGNA DISTRICT

Serial port

- Il codice qui accanto invia un messaggio di avviso nel caso in cui la temperatura superi i 30 gradi

```
void setup()
{
    Serial.begin(9600);
}

void loop()
{
    // codice...

    if (temp > 30)
    {

Serial.println(“WARNING!”);
    }

    // codice...
}
```

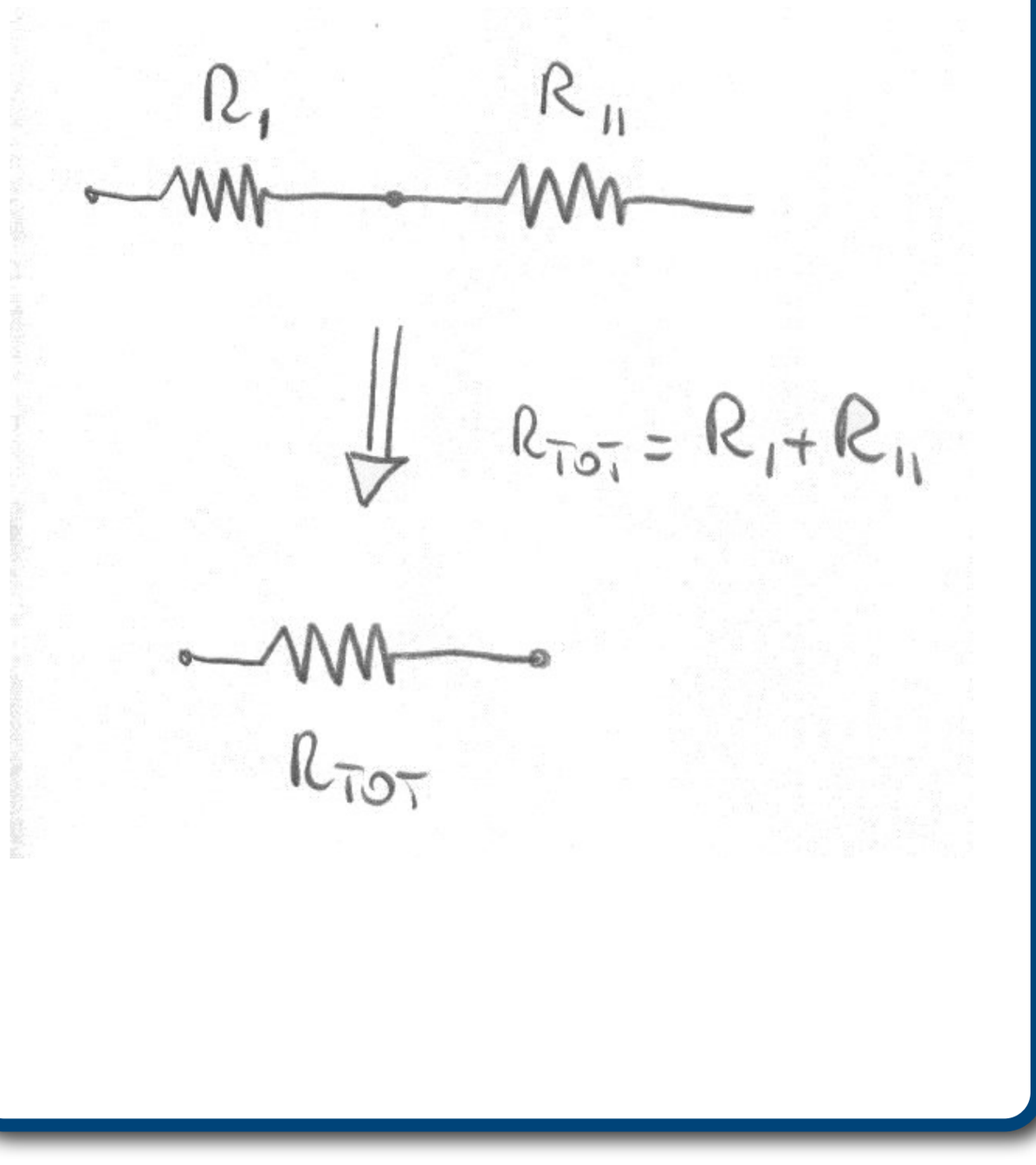
Resistenze variabili

- I sensori più semplici sono i resistori variabili, componenti passivi che modificano la propria resistenza in funzione di una azione meccanica, termica, luminosa...

Fotoresistore

Resistori in serie

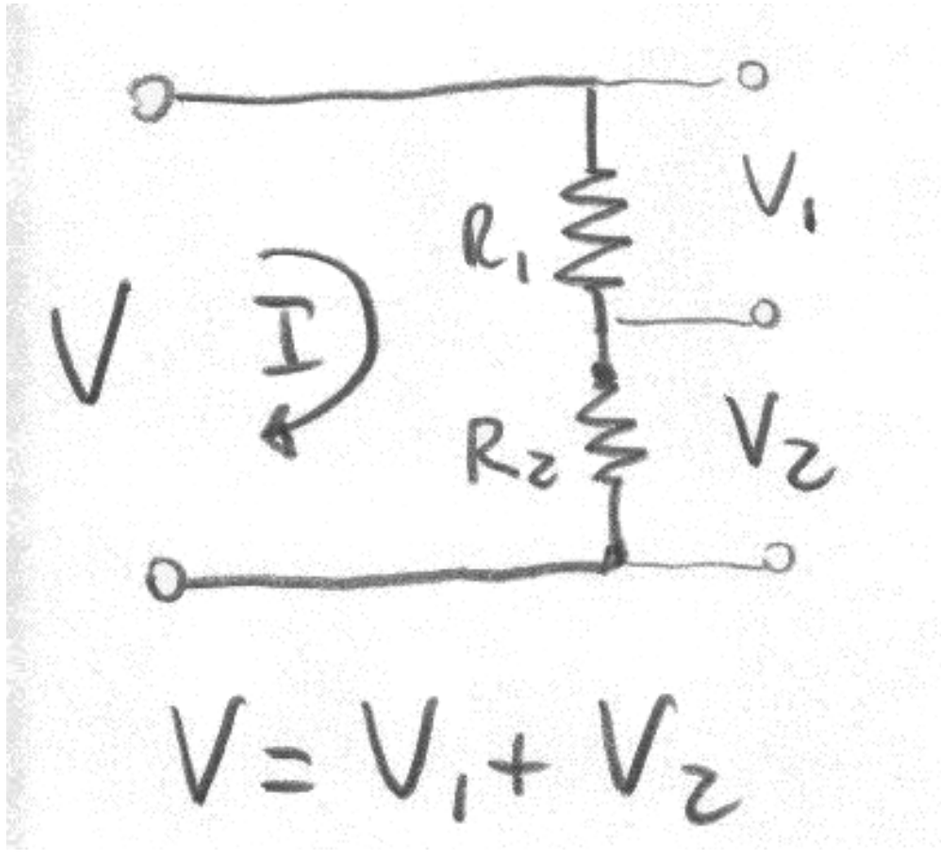
- Collegando due o più resistori **in serie**, la resistenza equivalente è pari alla **somma delle singole resistenze**



LAB Open Media Center
SARDEGNA DISTRICT

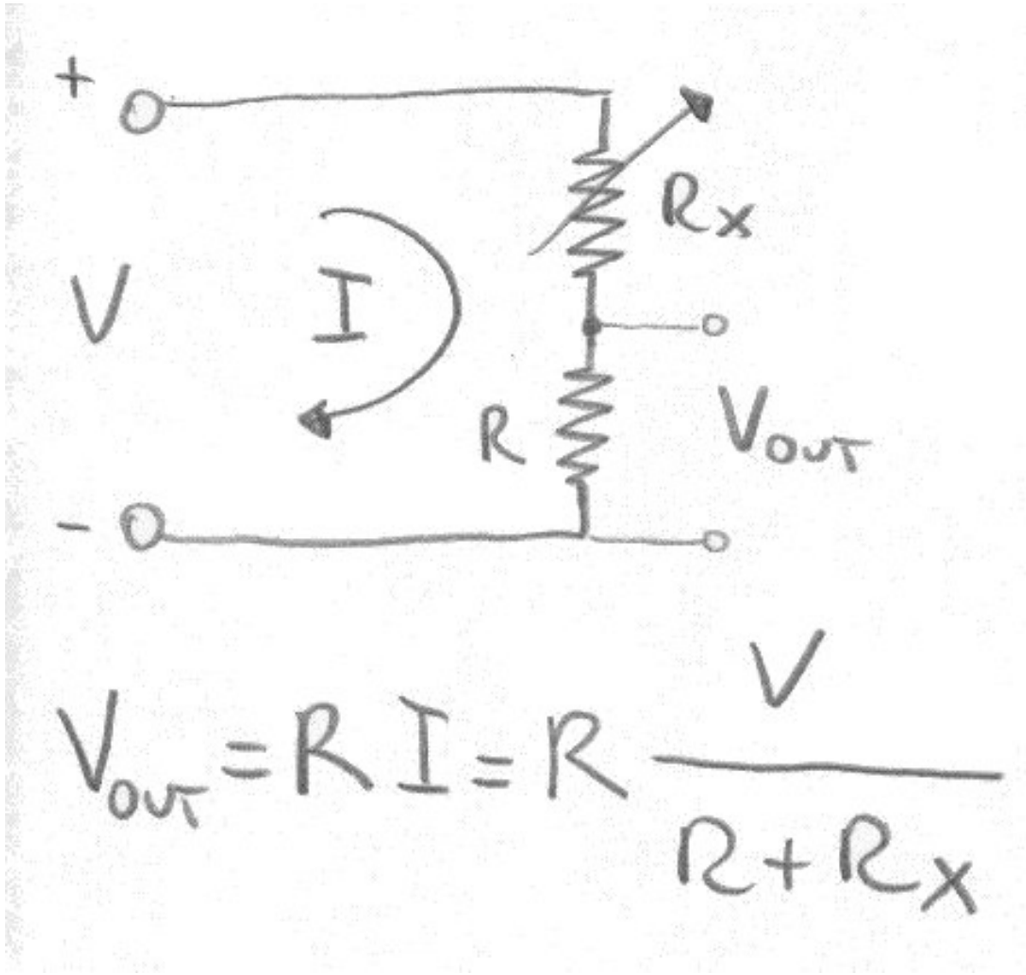
Partitore

- Due resistenze in serie costituiscono un partitore di tensione: la tensione di alimentazione ai capi del circuito è pari alla somma delle cadute di tensione ai capi di ciascuna resistenza



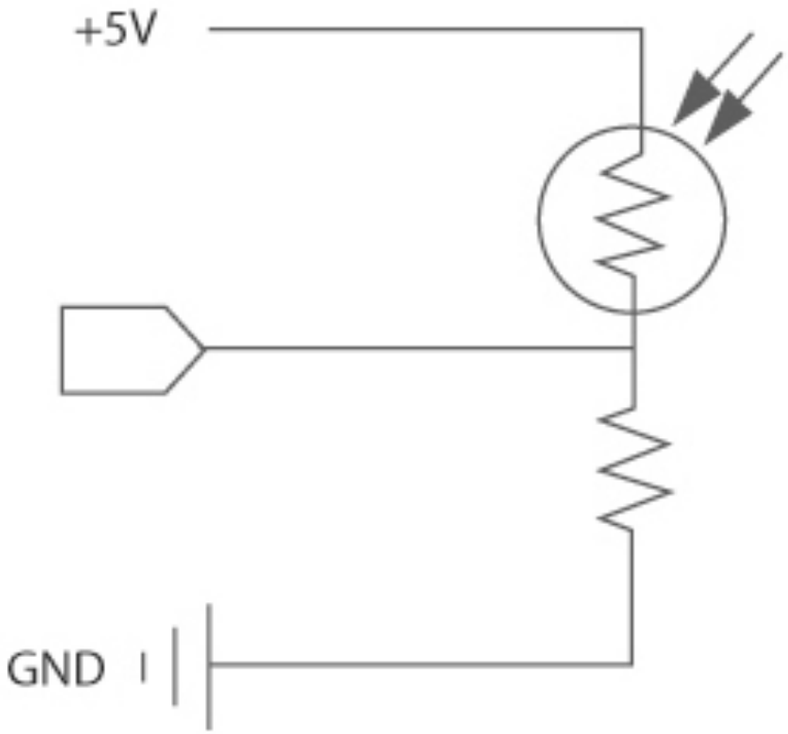
Partitore

- In un partitore, la una **resistenza variabile** provoca una **variazione di corrente** che, a sua volta, produce una **variazione di tensione** ai capi delle resistenze



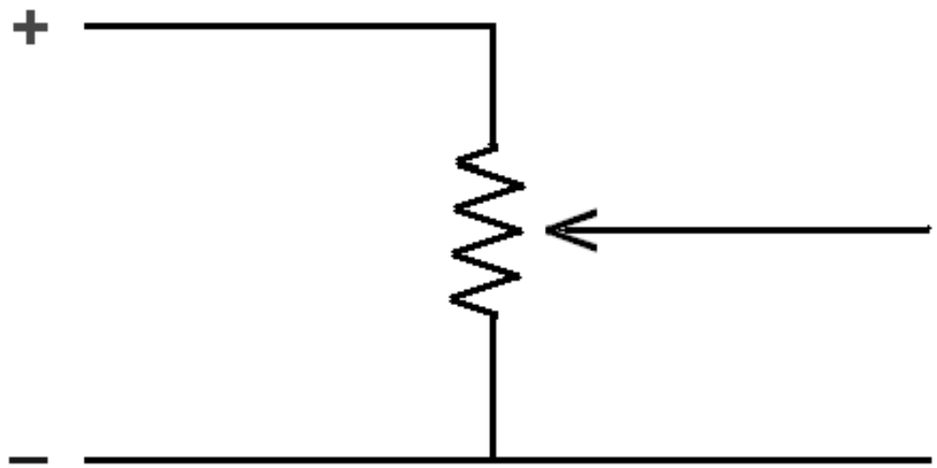
Fotoresistore

- Al variare dell'intensità luminosa varierà la resistenza del fotoresistore e, dunque, la tensione misurata ai capi della resistenza costante



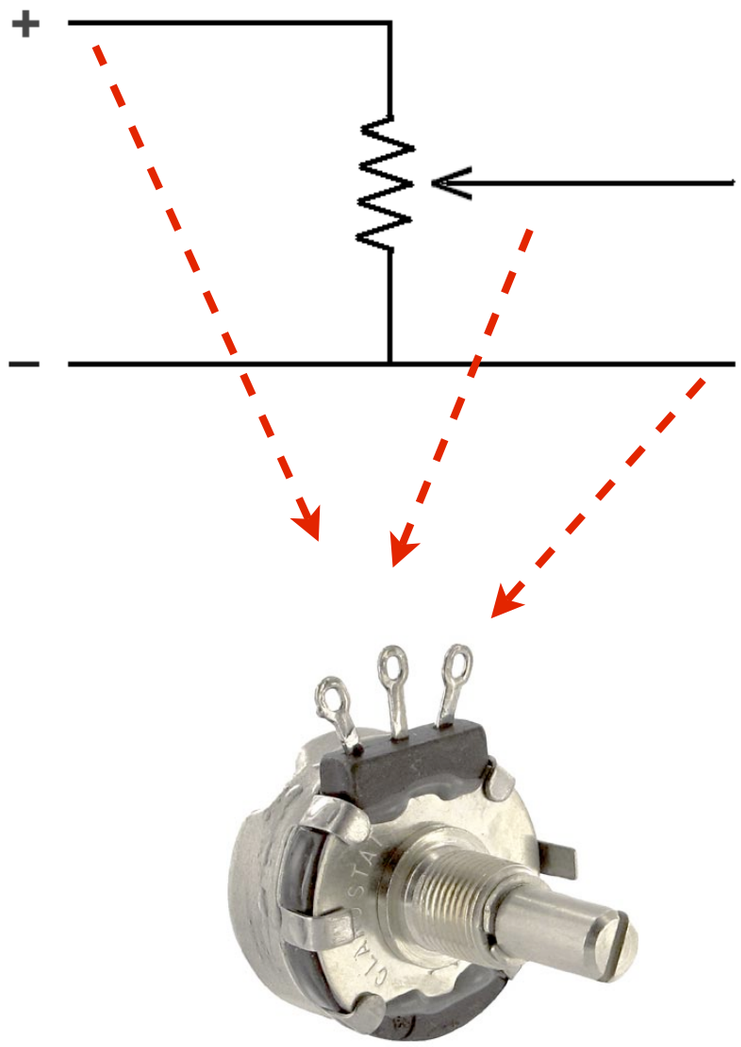
Potenziometro

- È un resistore la cui resistenza varia in base ad uno spostamento meccanico
- La lettura della tensione in uscita avviene tra il PIN centrale e la massa



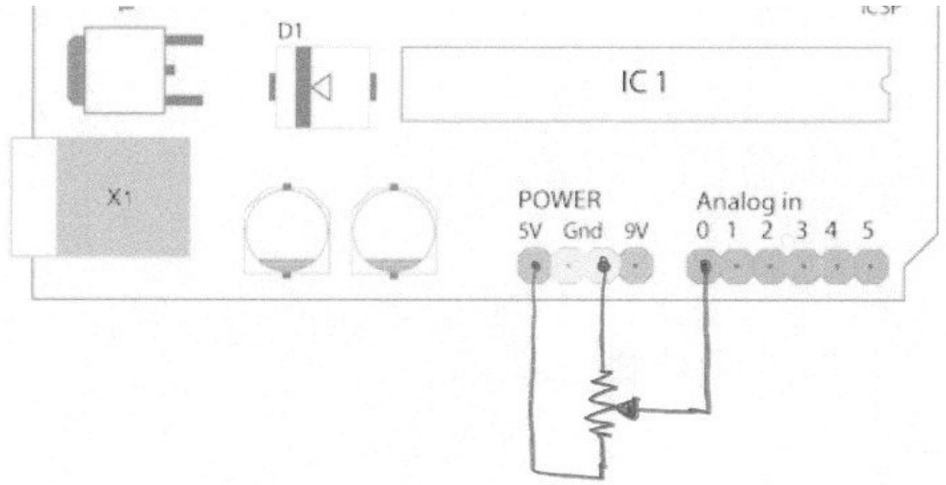
Potenziometro

- Ruotando in senso antiorario il cursore si sposta verso massa, riducendo il valore del secondo ramo resistivo
- Ruotando in senso orario il cursore si sposta verso la tensione di alimentazione, aumentando la resistenza del secondo ramo



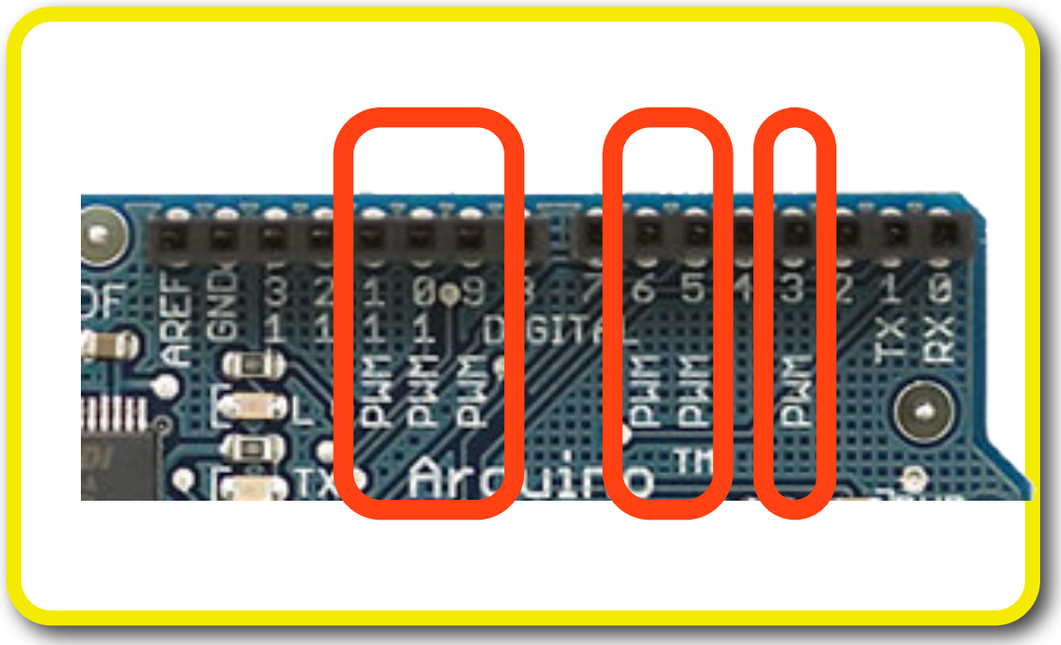
Potenziometro

- I contatti agli estremi saranno collegati alla tensione (5V)
- Il contatto centrale sarà connesso ad un ingresso analogico: la lettura del valore di tensione avviene con analogRead()



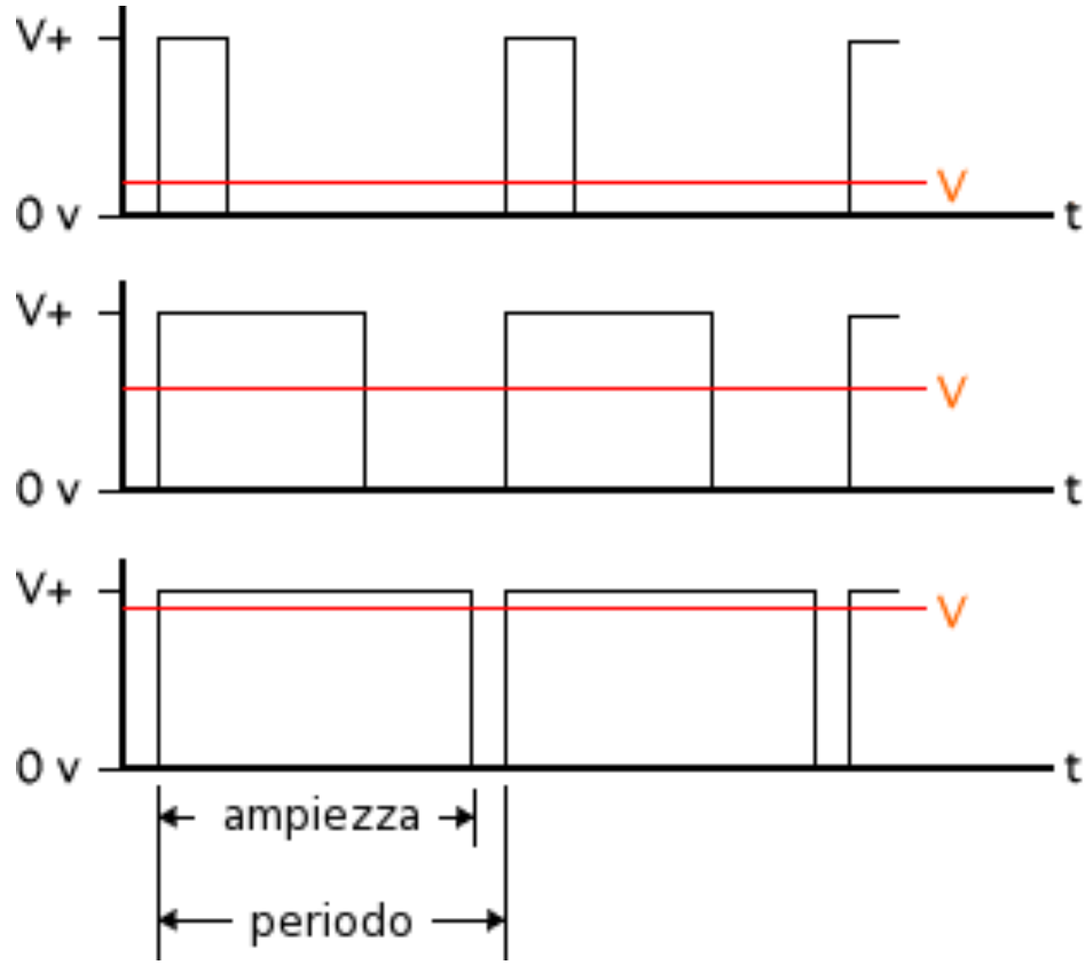
Output analogico (PWM)

- È possibile attivare un output analogico sotto forma di treno di impulsi di ampiezza variabile (Pulse Width Modulation, PWM)
- La funzione analogWrite() riceve come parametri il PIN di uscita e l'ampiezza temporale dell'impulso nel range 0-255
- L'output PWM è disponibile solamente sui PIN 3, 5 ,6 9, 10 e 11



Output “analogico”

- La frequenza degli impulsi è circa 490 Hz
- Maggiore è la durata dell’impulso e maggiore sarà la tensione media sul periodo



Fade in & out [I]

- Utilizzando un PIN PWM è possibile variare l'intensità (media sul periodo) dell'uscita analogica per controllare la luminosità di una luce, l'intensità di un suono o la velocità di un motore

```
int LED_PIN = 3;
int w = 0;

void setup()
{
  /*
  non e' necessario impostare
  il pinMode in OUTPUT,
pero'...
  */
}
```

Fade in & out [2]

```
void loop()
{
    for(w = 0 ; w <= 255; w+=5)
    {
        analogWrite(LED_PIN, w);
        delay(50);
    }

    for(w = 255; w >=0; w-=5)
    {
        analogWrite(LED_PIN, w);
        delay(50);
    }
}
```

Aumenta l'ampiezza dell'impulso finché non raggiunge il massimo. Ad ogni ciclo attende 50 ms prima dell'incremento successivo

Decrementa l'ampiezza dell'impulso finché non è nulla. Ad ogni ciclo attende 50 ms prima dell'iterazione successiva

LAB Open MediaCenter
SARDEGNA DISTRICT

Dimmer [I]

- Il dimmer è un circuito che consente di regolare la l'intensità di una sorgente luminosa

```
int LED_PIN = 3;
int DIMMER_PIN =0;

// memorizza il valore
precedente
int lastValue = 0;

// memorizza l'ultimo valore
letto
int value;

void setup()
{
}
```

Dimmer [2]

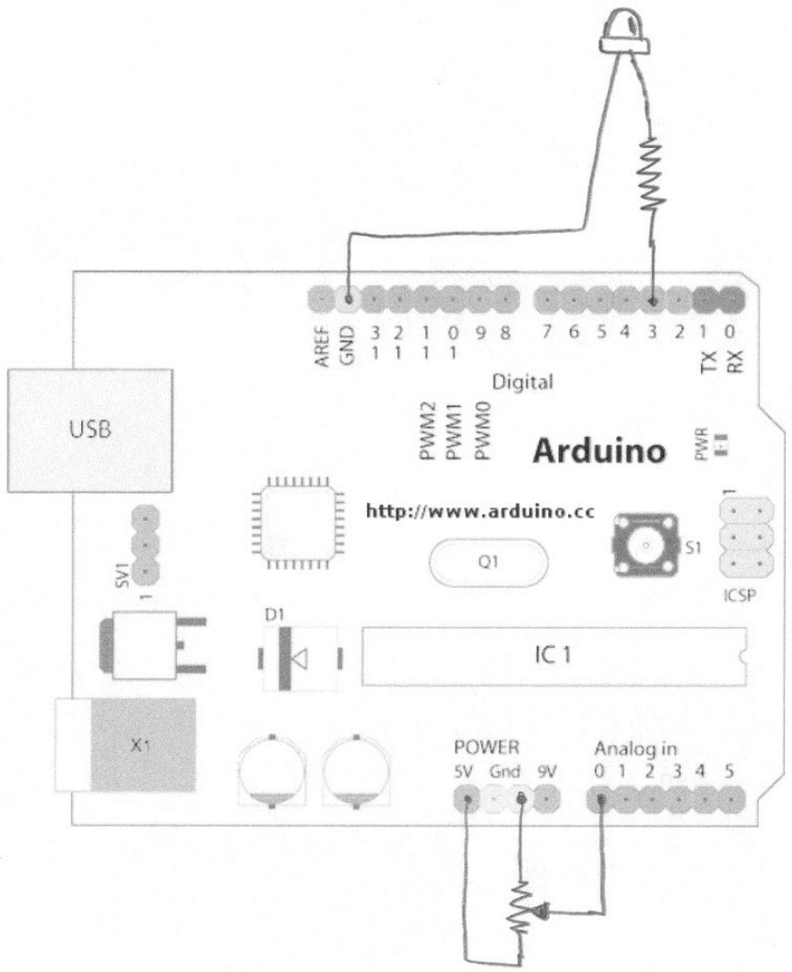
- Utilizzando un potenziometro e un LED è possibile realizzare un semplice dimmer... per la casa delle bambole!

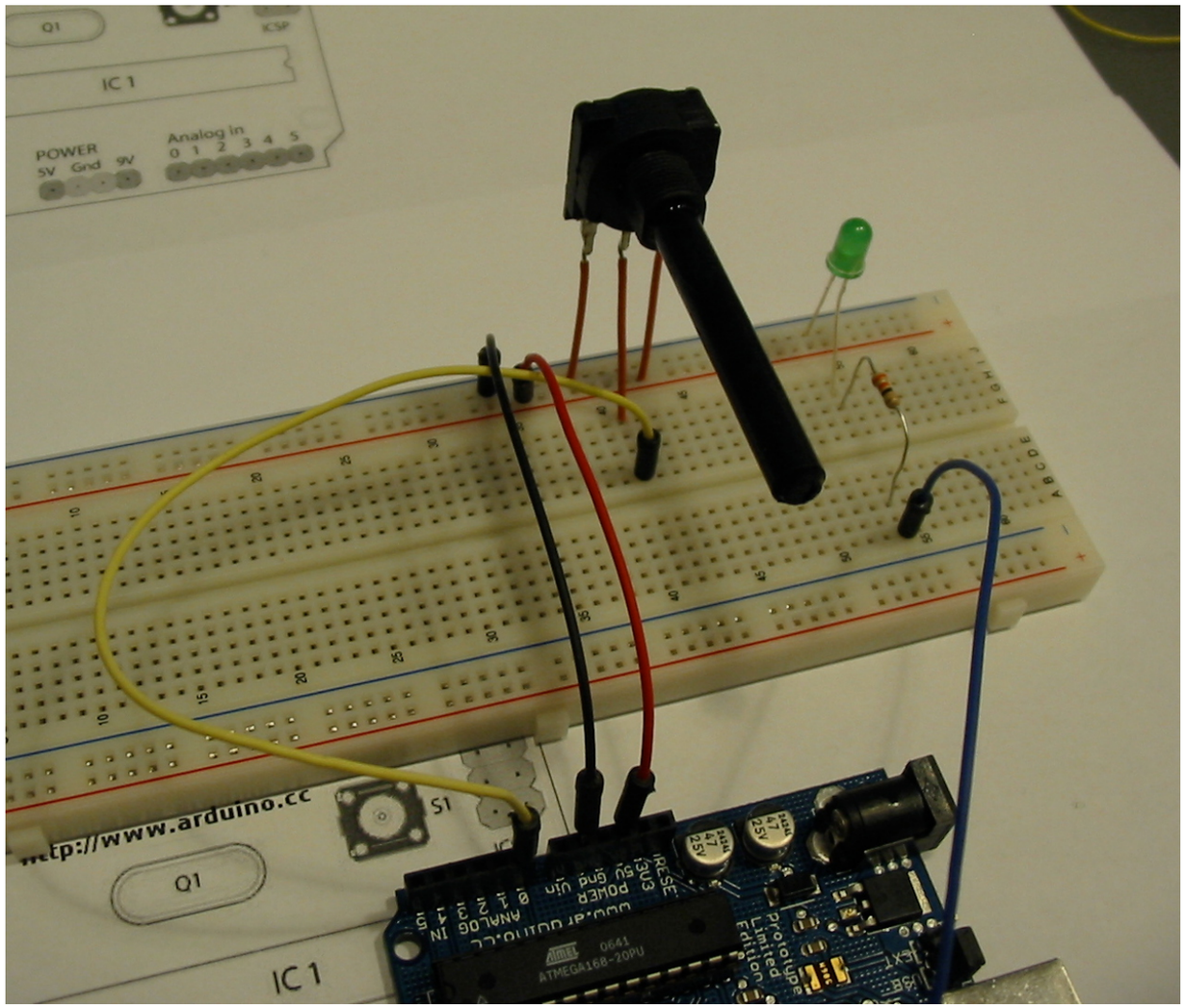
```
void loop()
{
  value =
analogRead(DIMMER_PIN) / 4;

  if (value != lastValue)
  {
    lastValue = value;

    analogWrite(LED_PIN,
value);
    }
}
```

LAB Open MediaCenter

SARDEGNA DISTRICT

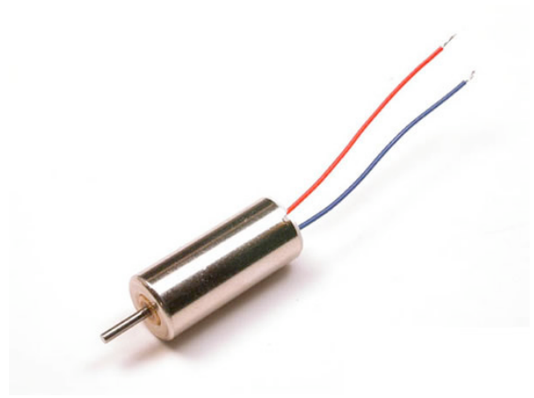
Dimmer [3]

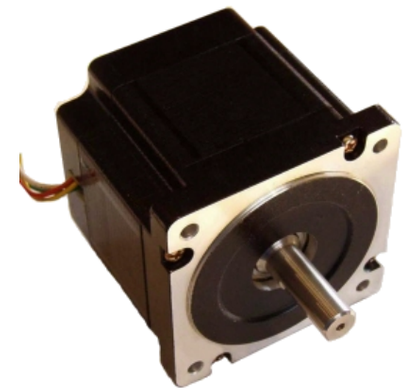




Motori

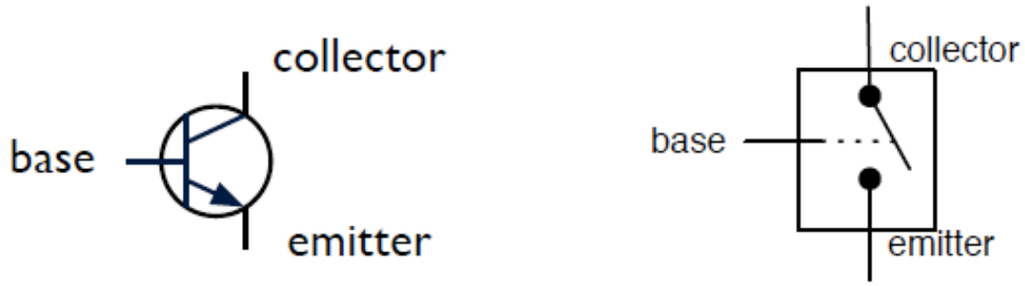
- Arduino consente il controllo dei tre tipi di motori elettrici:
  - motori diretti
  - servomotori
  - motori passo-passo
- Arduino **non alimenta** i motori ma ne controlla il movimento

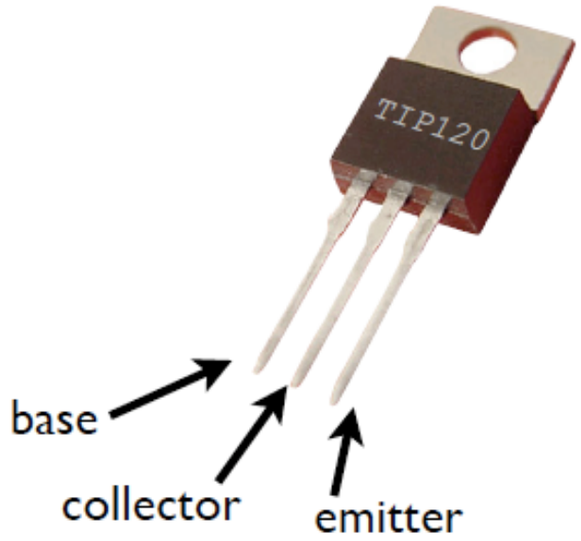




Transistor

- È un componente a semiconduttore utilizzato come interruttore controllato
- Polarizzando la **base**, **collettore** ed **emettitore** conducono corrente, comportandosi come un circuito chiuso

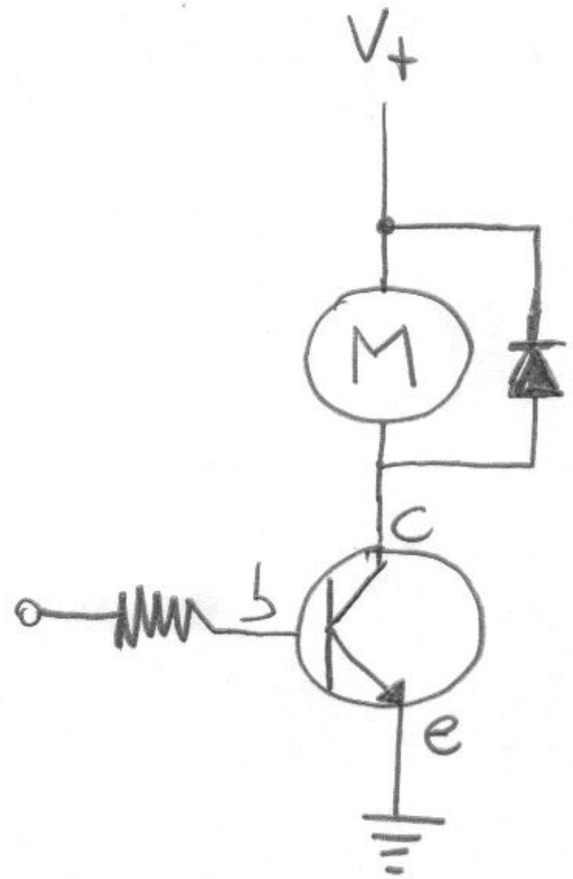




Controllo motore

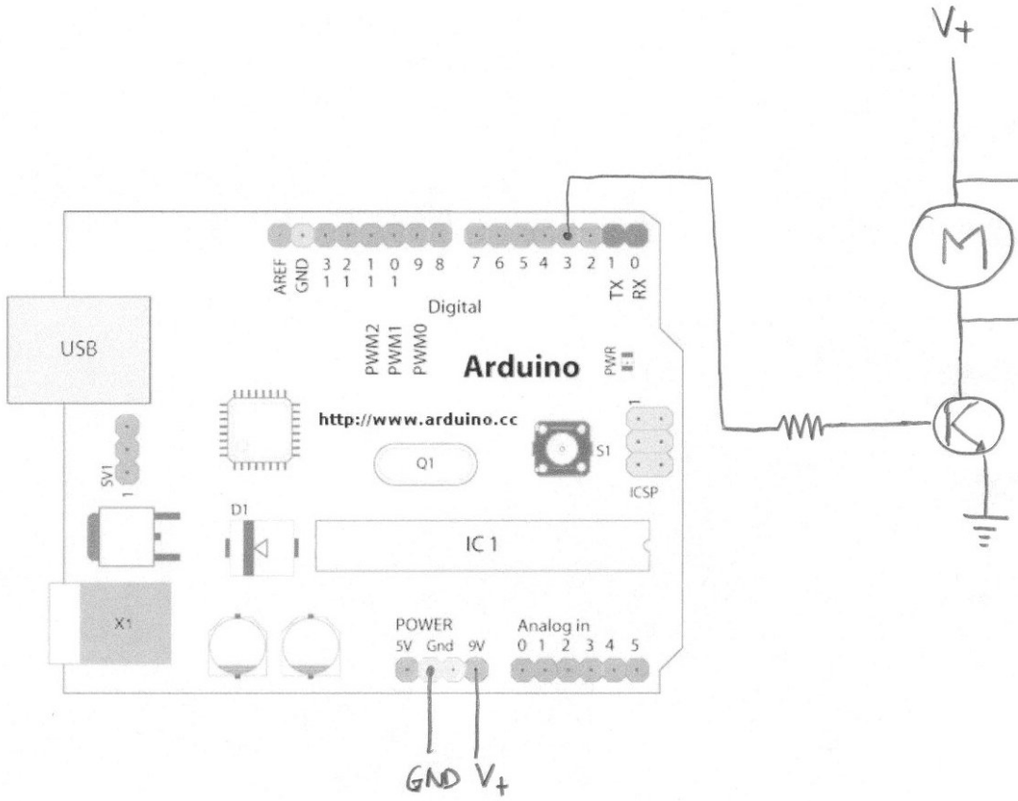
- Portando a livello alto la base, il transistor conduce e alimenta il motore

- Il diodo in parallelo elimina la tensione indotta dal motore

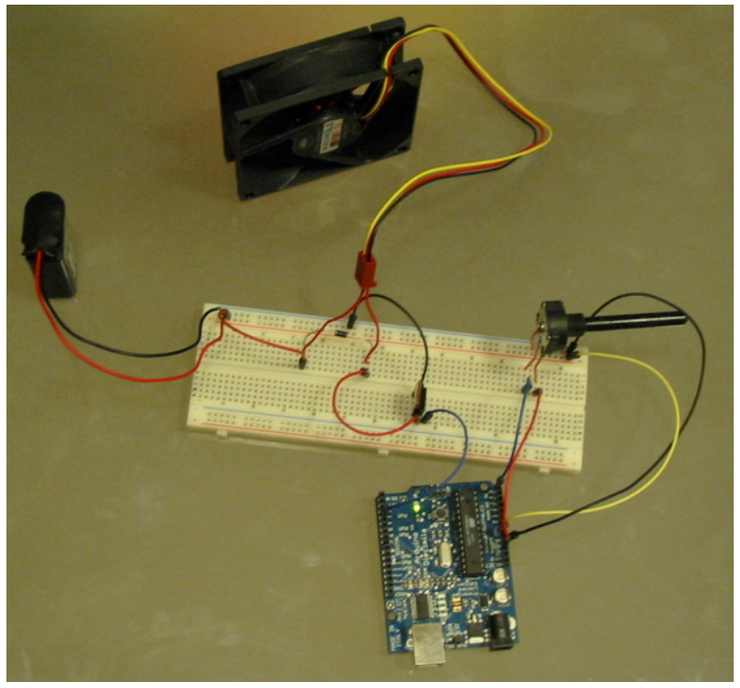


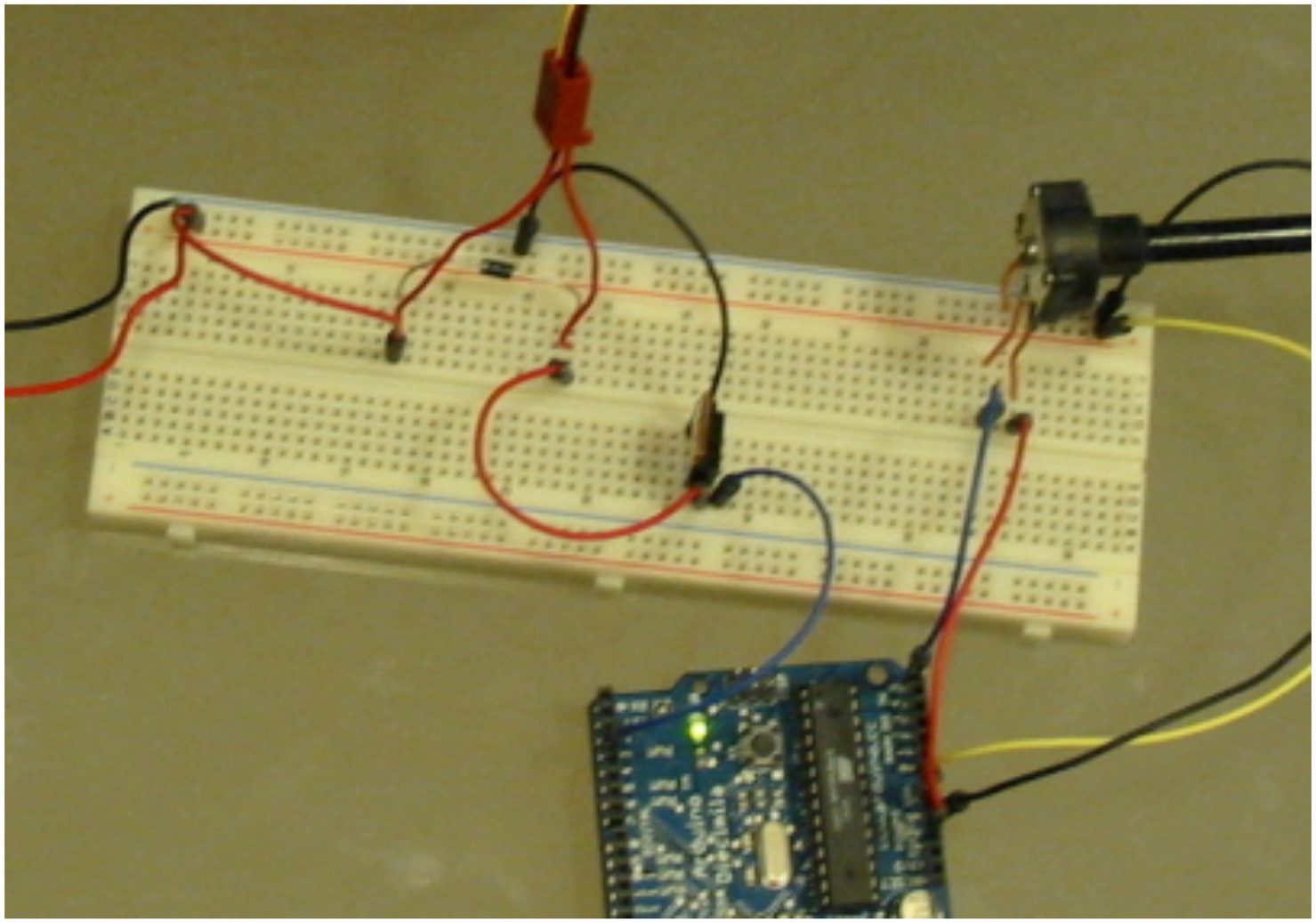
Controllo PWM

- Utilizzando una uscita analogica è possibile variare la velocità di rotazione del motore modificando l'ampiezza dell'impulso



Un motore controllato

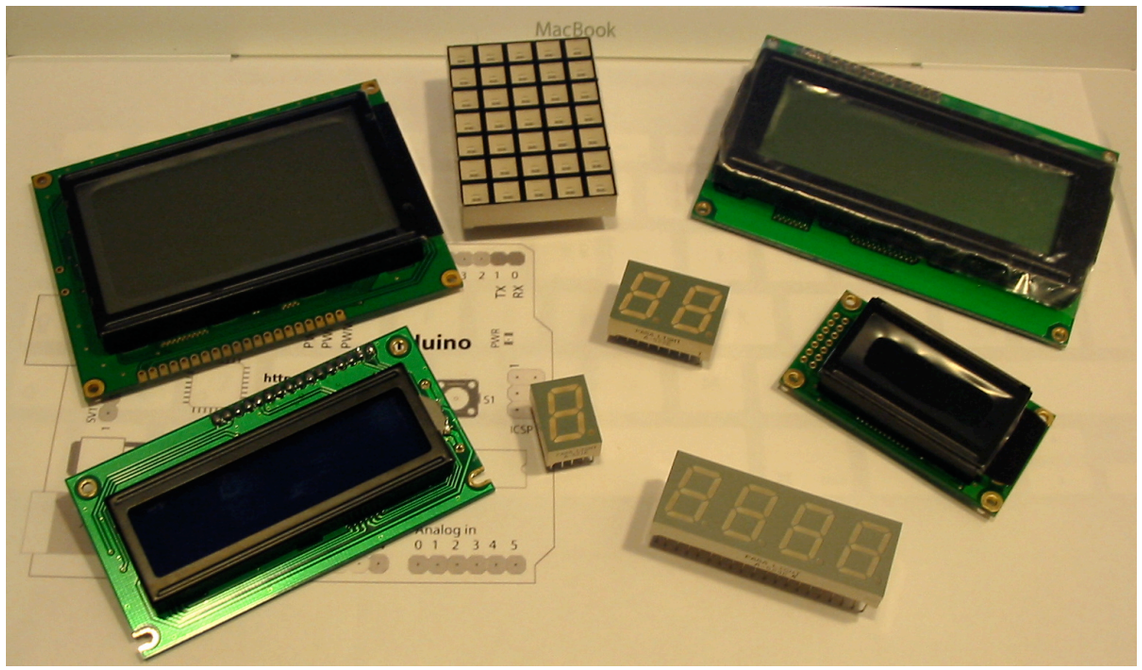




LAB Open Media Center
SARDEGNA DISTRICT

Display

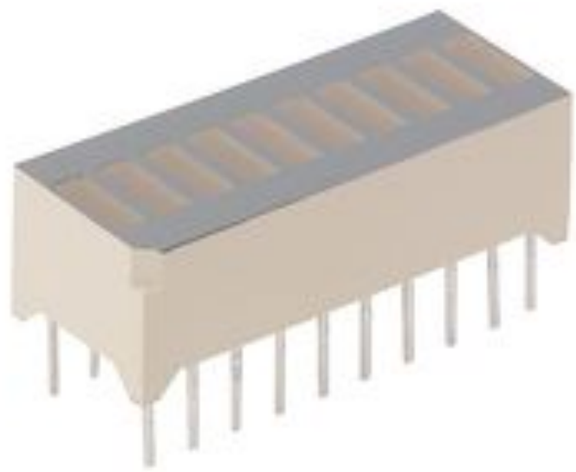
- I display consentono ad Arduino di visualizzare informazioni complesse: intensità, numeri, testi e grafica:
  - BAR graph
  - cifre LED a 7 segmenti
  - LCD alfanumerici
  - LCD grafici



LAB Open Media Center
SARDEGNA DISTRICT

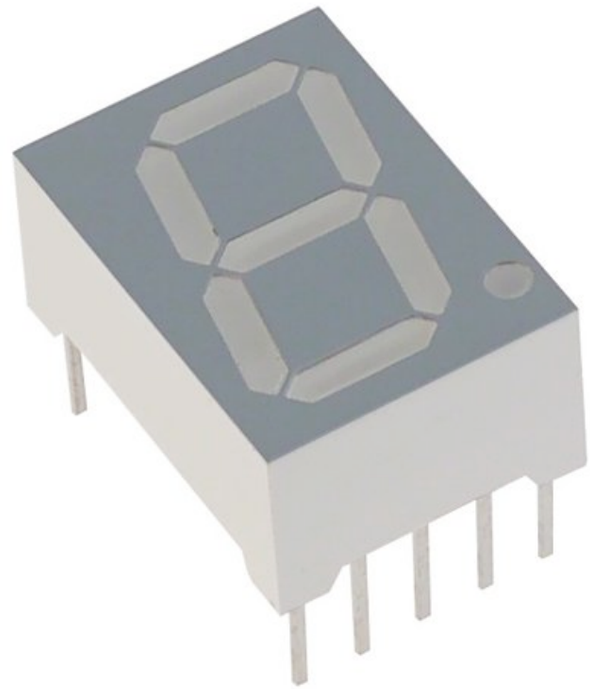
BAR Graph

- È il display più semplice: è costituito da una sequenza ordinata di LED che possono essere accesi in successione per indicare un livello



Display 7 segmenti

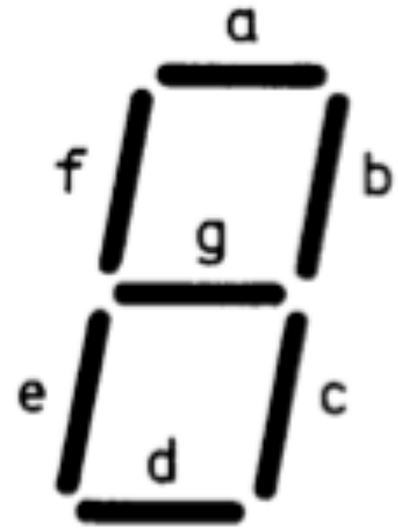
- Due configurazioni:
  - Anodo comune: un unico PIN è connesso a V+, il controllo avviene sul catodo (LOW) di ciascun LED
  - Catodo comune: un unico PIN è connesso a GND, il controllo avviene sull'anodo (HIGH) di ciascun LED



LAB Open Media Center
SARDEGNA DISTRICT

Display 7 segmenti

- Questo tipo di display richiede 7 PIN digitali, uno per ciascun LED
- È possibile ridurre gli output a 4 utilizzando un display driver e la codifica BCD



Da decimale a binario

- Da base decimale a base binaria:
  - si divide il numero per 2 ed il resto rappresenta il bit meno significativo del numero decimale
  - il quoziente è diviso nuovamente per 2, e il resto sarà la cifra decimale successiva
  - si continua con finché il quoziente è nullo

```
Conversione di 87 in binario:

87 / 2 = 43   ->    87 % 2 = 1
43 / 2 = 21   ->    43 % 2 = 1
21 / 2 = 10   ->    21 % 2 = 1
10 / 2 =  5   ->    10 % 2 = 0
 5 / 2 =  2   ->     5 % 2 = 1
 2 / 2 =  1   ->     2 % 2 = 0
 1 / 2 =  0   ->     1 % 2 = 1

87 in binario diventa: 1010111
```

Da binario a decimale

- La conversione di un numero da base binaria a base decimale si effettua attraverso la sommatoria del prodotto di ciascuna cifra per 2 elevato il peso (la "posizione", da 0 a N-1) della cifra stessa

Conversione di 1010111 in decimale:

```
1  ->  1 * 2^6 = 64
0  ->  0 * 2^5 =  0
1  ->  1 * 2^4 = 16
0  ->  0 * 2^3 =  0
1  ->  1 * 2^2 =  4
1  ->  1 * 2^1 =  2
1  ->  1 * 2^0 =  1
```

64 + 16 + 4 + 2 + 1 = 87

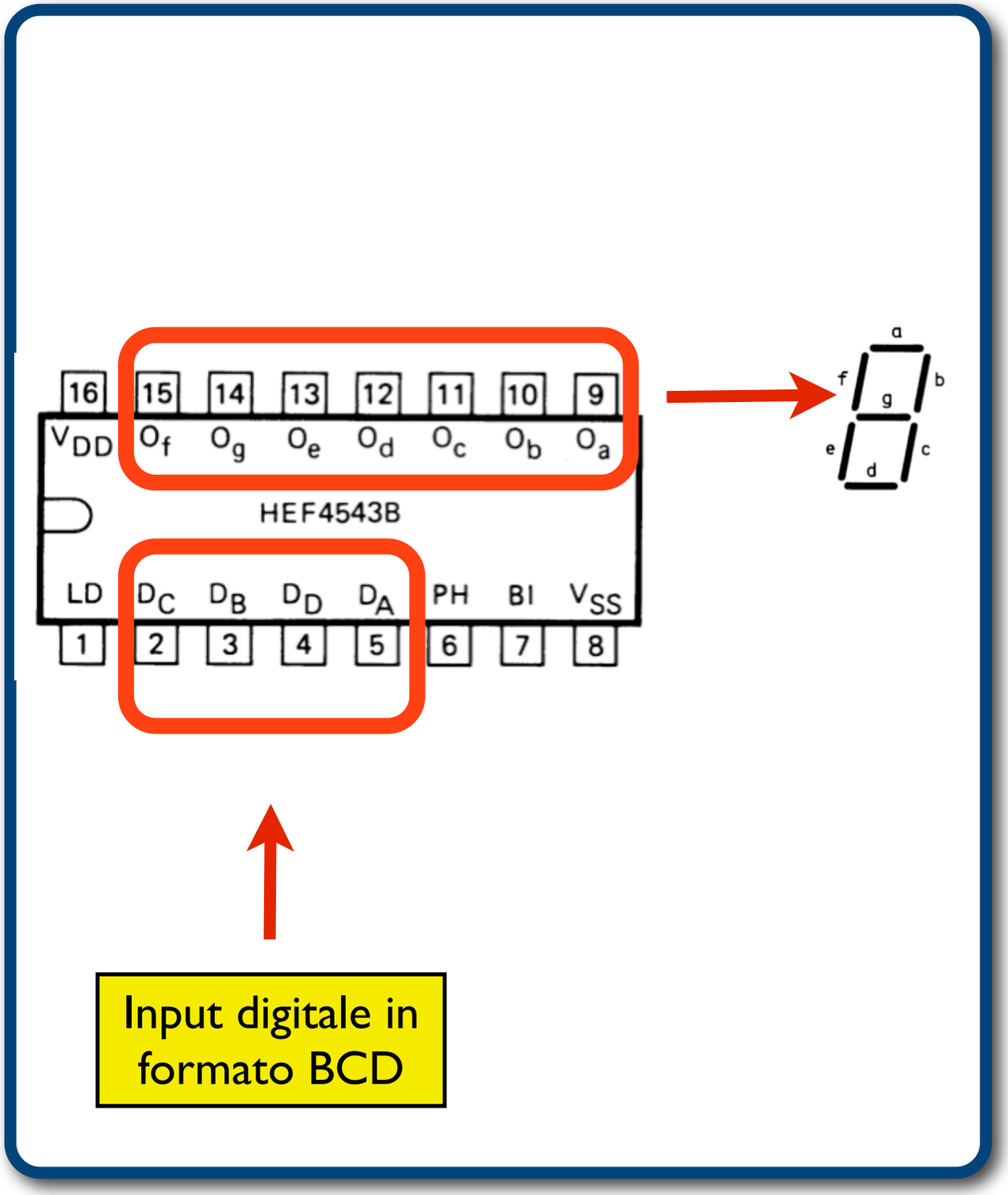
BinaryCodedDecimal

- Il BCD è un formato di codifica di numeri decimali in binario, la cui trasformazioni avviene in maniera indipedente per ciascuna cifra
- Il numero risultante deve essere letto a gruppi di 4 bit per volta

```
Codifica binaria:

12     ->    1100
123    ->    01111011


Codifica BCD:

12  ->    1 2 -> 0001 0010
123 -> 1 2 3 -> 0001 0010 0011
```

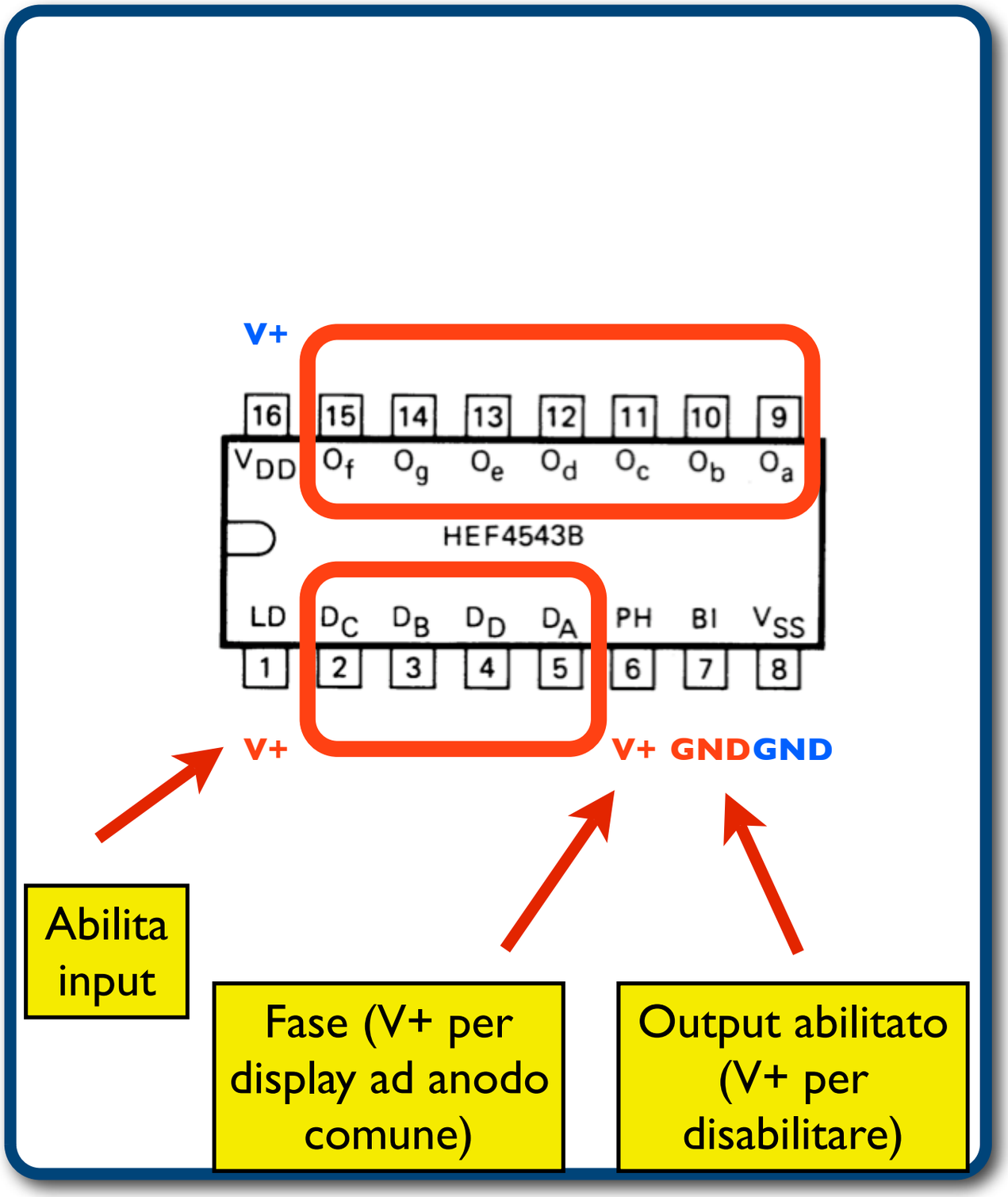
Driver HEF 4543

- Il driver HEF4543 è alimentato a 3V+
- I PIN DA, DB, DC, DD ricevono l'input della cifra decimale in formato BCD
- I PIN Oa - Og controllano i segmenti del display a LED



Driver HEF 4543

- Configurazione:
  - **LD**: se LOW, disabilita l'input e "congela" l'output
  - **PH**: se LOW consente di pilotare display a **catodo comune**; HIGH per display ad **anodo comune**
- **BI**: impostato HIGH disabilita l'output (il display si spegne)



LAB Open Media Center
SARDEGNA DISTRICT

Un contatore

```
int D[] = {5, 4, 3, 2};

int L[] = { 0, 0, 0, 0,
            0, 0, 0, 1,
            0, 0, 1, 0,
            0, 0, 1, 1,
            0, 1, 0, 0,
            0, 1, 0, 1,
            0, 1, 1, 0,
            0, 1, 1, 1,
            1, 0, 0, 0,
            1, 0, 0, 1 };

int j;
int i;
```

PIN Output

0000 = 0
0001 = 1
0010 = 2
0011 = 3
0100 = 4
0101 = 5
0110 = 6
0111 = 7
1000 = 8
1001 = 9

LAB Open Media Center
SARDEGNA DISTRICT

Un contatore

```
void setup()
{
  for (i = 0; i < 4; i++)
  {
    pinMode(D[i], OUTPUT);
  }
}
```

loop()

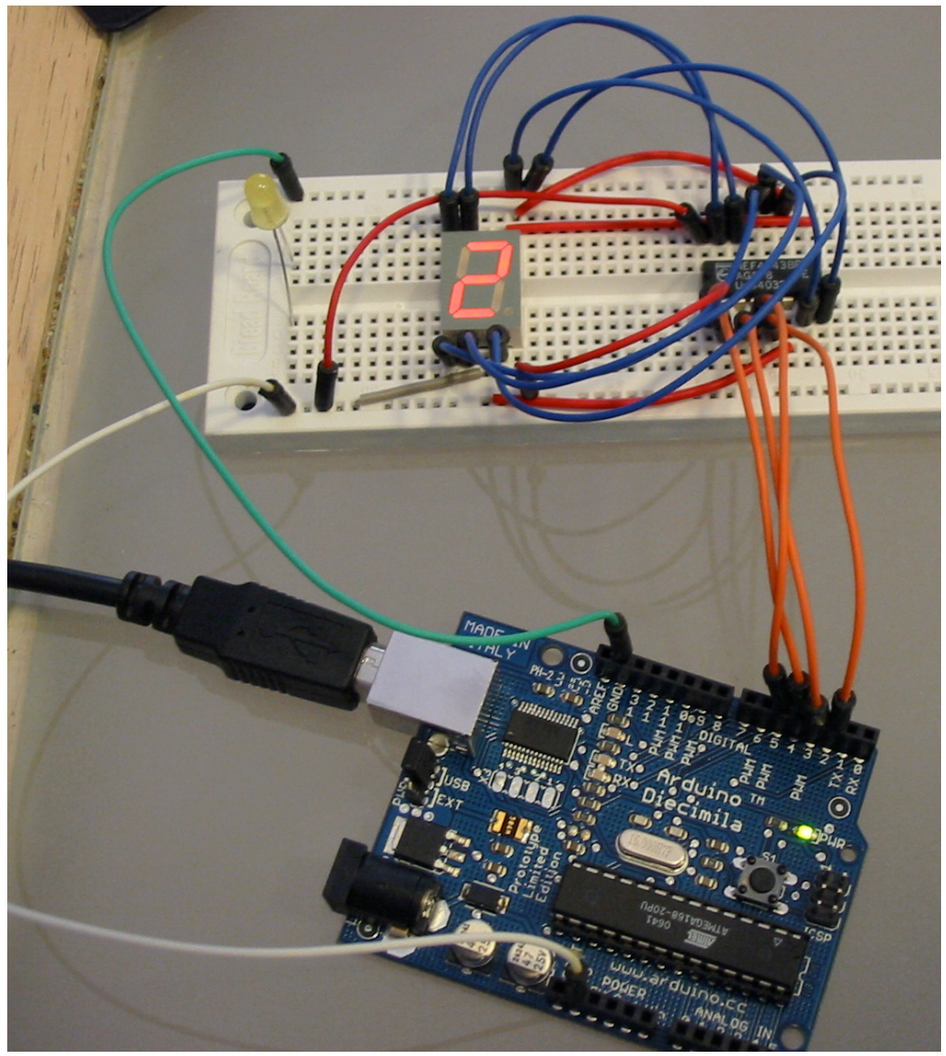
- Il ciclo esterno scorre le cifre decimali (le righe del vettore L)
- Il ciclo interno scrive l'output digitale della codifica BCD di ciascuna cifra decimale

```
void loop()
{
  for (i = 0; i <  10; i++)
  {
    for (j = 0; j < 4; j++)
    {
      digitalWrite(5-j, L[j+4*i]);
    }

    delay(1000);
  }
}
```

LAB Open MediaCenter
SARDEGNA DISTRICT

Prototipo

- La breadboard è piuttosto affollata attorno al display driver, ma il collegamento ad Arduino è ordinato e soprattutto molti PIN sono liberi per rivere input o controllare altri dispositivi



Hands-on project

Data Watcher

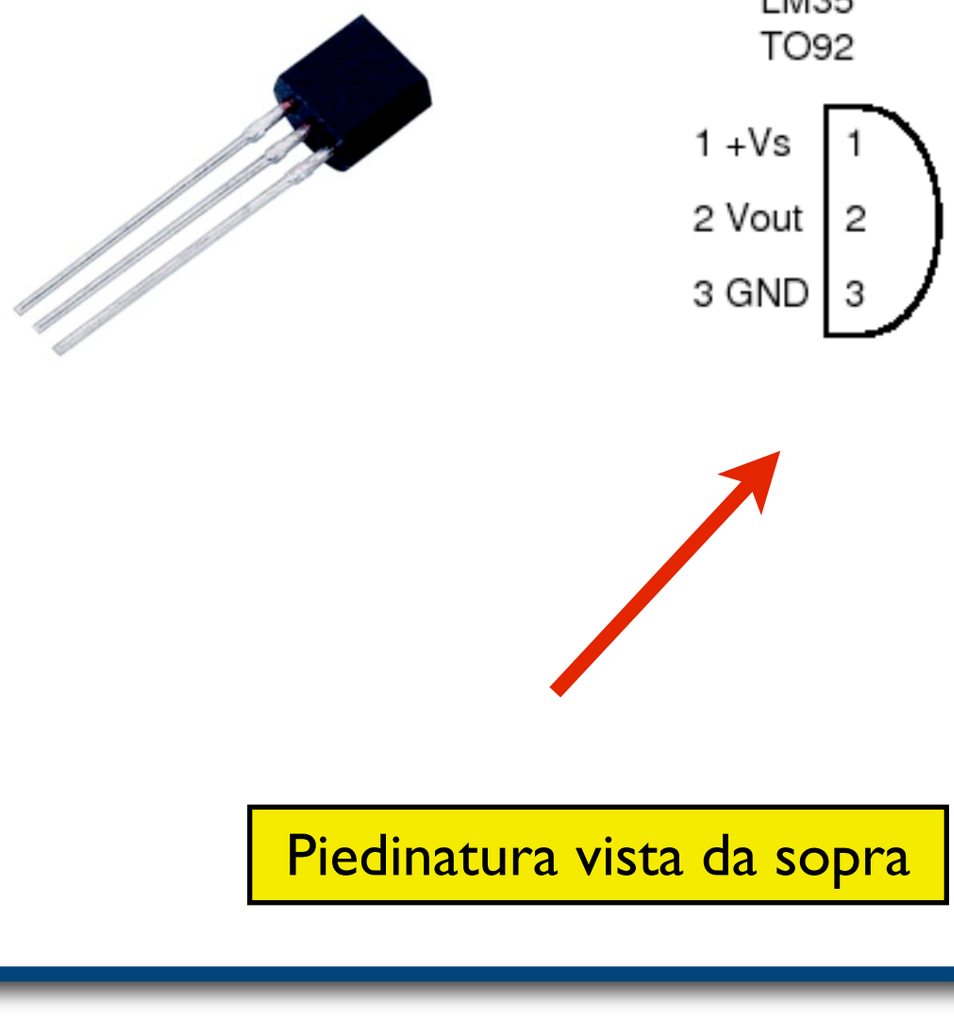
LAB Open Media Center

SARDEGNA DISTRICT

CRS4

Sensore termico

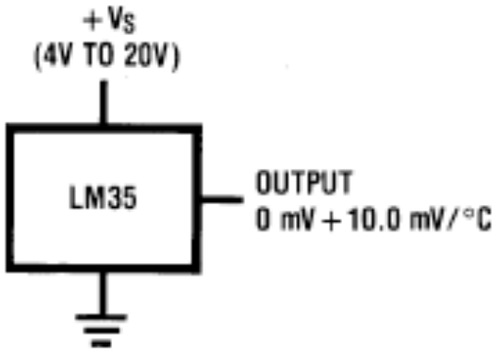
- Il sensore attivo LM35DZ restituisce una tensione d'uscita proporzionale alla temperatura ambietale



Piedinatura vista da sopra

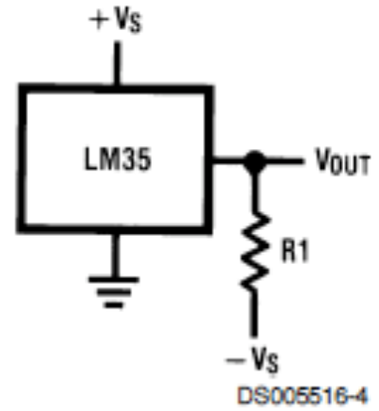
LAB Open Media Center
SARDEGNA DISTRICT

LM35DZ



DS005516-3

FIGURE 1. Basic Centigrade Temperature Sensor (+2˚C to +150˚C)



DS005516-4

Choose $R_1 = -V_S/50\ \mu A$

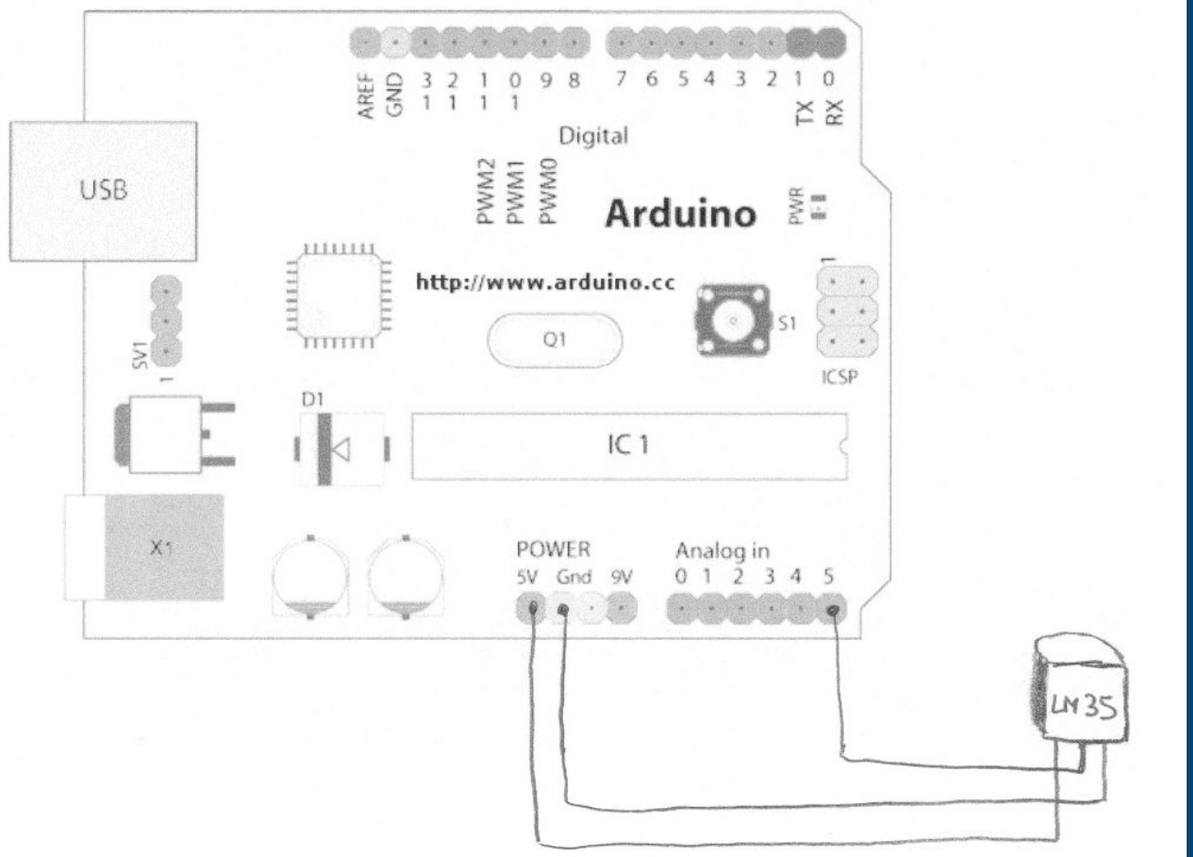
V_{OUT} = +1,500 mV at +150˚C
= +250 mV at +25˚C
= −550 mV at −55˚C

FIGURE 2. Full-Range Centigrade Temperature Sensor

http://www.national.com/ds/LM/LM35.pdf

Il circuito

- Arduino può alimentare direttamente il sensore di temperatura (5V) e leggerne l'output attraverso un ingresso analogico

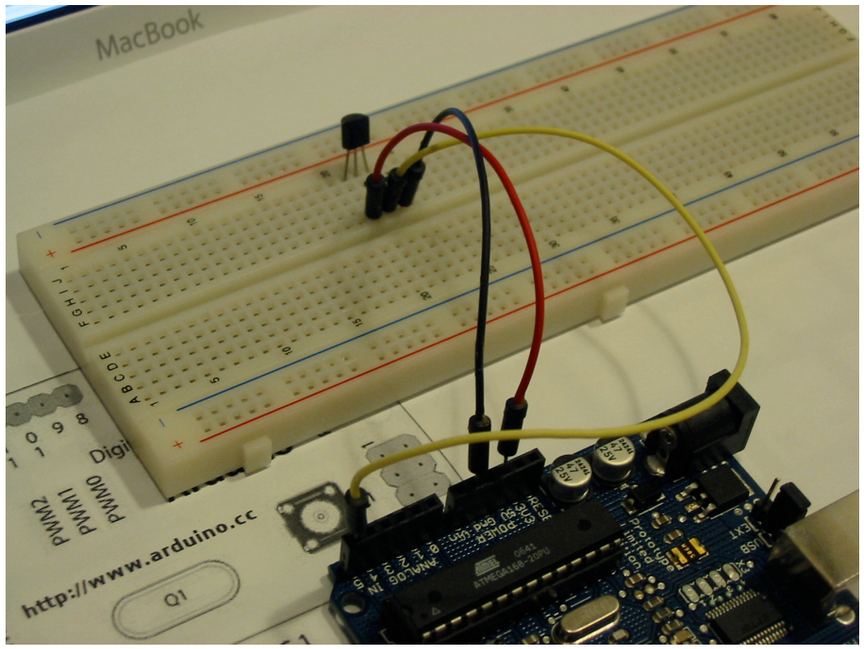


LAB Open MediaCenter
SARDEGNA DISTRICT

Il prototipo

```
int TEMP_PIN = 5;
int temp;

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    temp = 2 + analogRead(TEMP_PIN) * 0.5;

    Serial.println(temp, DEC);
    delay(100);
}
```

Ogni step di analogRead() corrisponde a 5mv (circa); un grado corrisponde ad un intervallo di 10mV, dunque:
analogRead(TEMP_PIN) * 5 / 10;
Nota: a tensione nulla corrisponde una temperatura di 2°

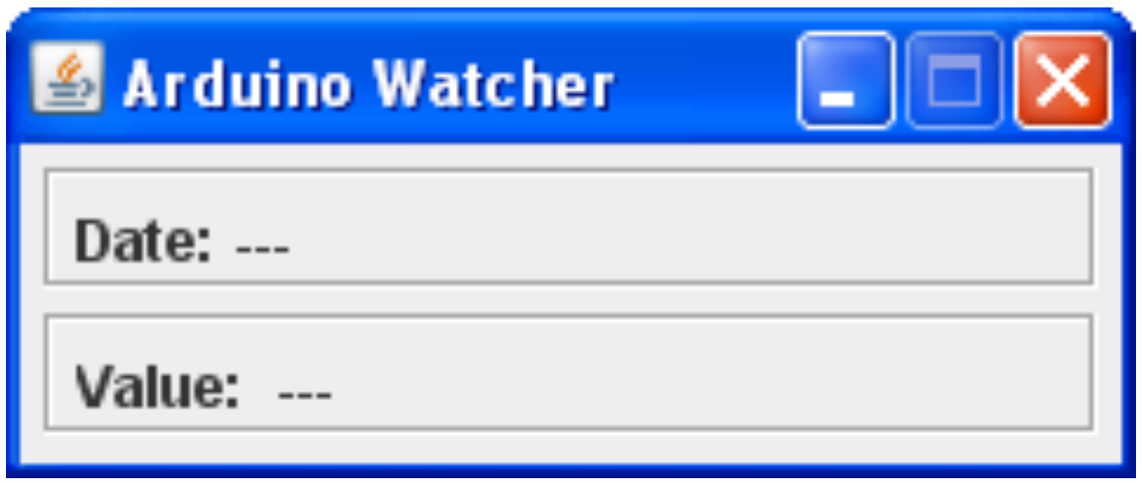


LAB Open Media Center
SARDEGNA DISTRICT

Arduino Watcher

- La temperatura è inviata sulla seriale come intero: è sufficiente aprire la porta seriale lato PC per leggere il valore rilevato dal sensore

- Utilizzando JSE e JavaComm è possibile realizzare un piccolo “Arduino Watcher” con cui tenere d’occhio la temperatura nell’ambiente



Hands-on project

Sensore wireless remoto

LAB Open MediaCenter

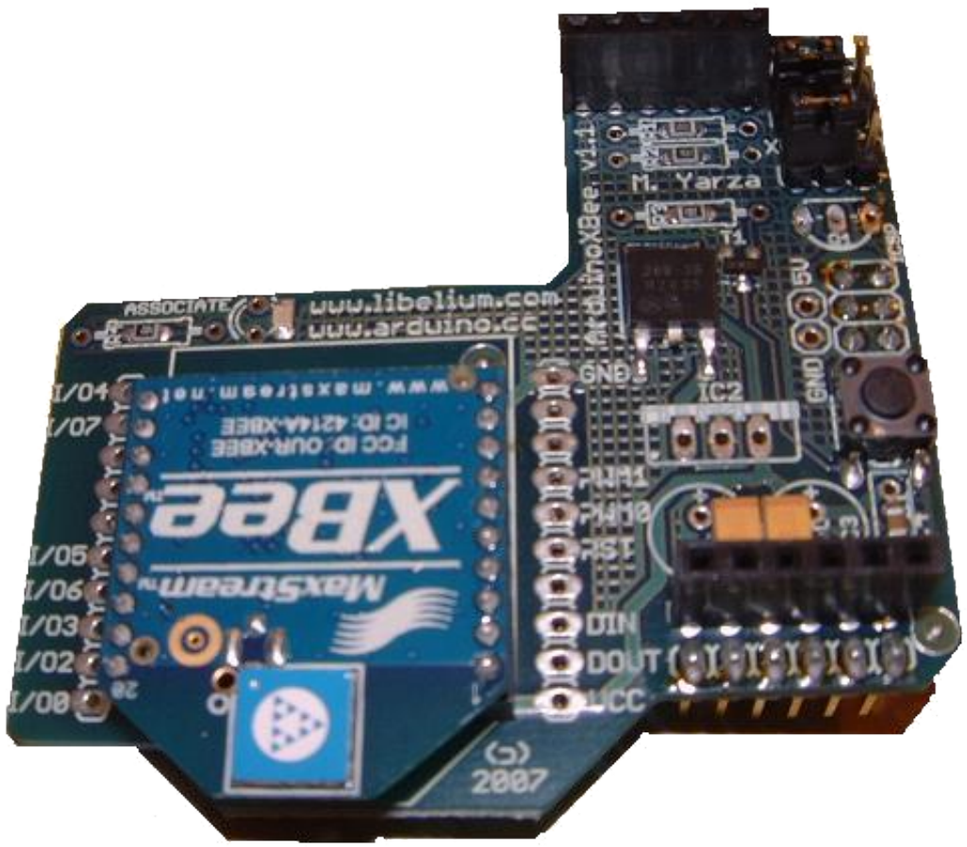
SARDEGNA DISTRICT

CRS4

LAB Open Media Center
SARDEGNA DISTRICT

Arduino XBee

- L'Arduino XBee Shield è un modulo aggiuntivo per Arduino che consente la comunicazione seriale attraverso il protocollo ZigBee



LAB Open Media Center
SARDEGNA DISTRICT

Wireless Sensor

- Utilizzando opportuni moduli radio connessi alla porta seriale di Arduino è possibile realizzare sensori wireless economici ed efficienti

- Non occorrono tecnologie dedicate: è sufficiente un qualsiasi modem radio con ingressi/uscite seriali TTL

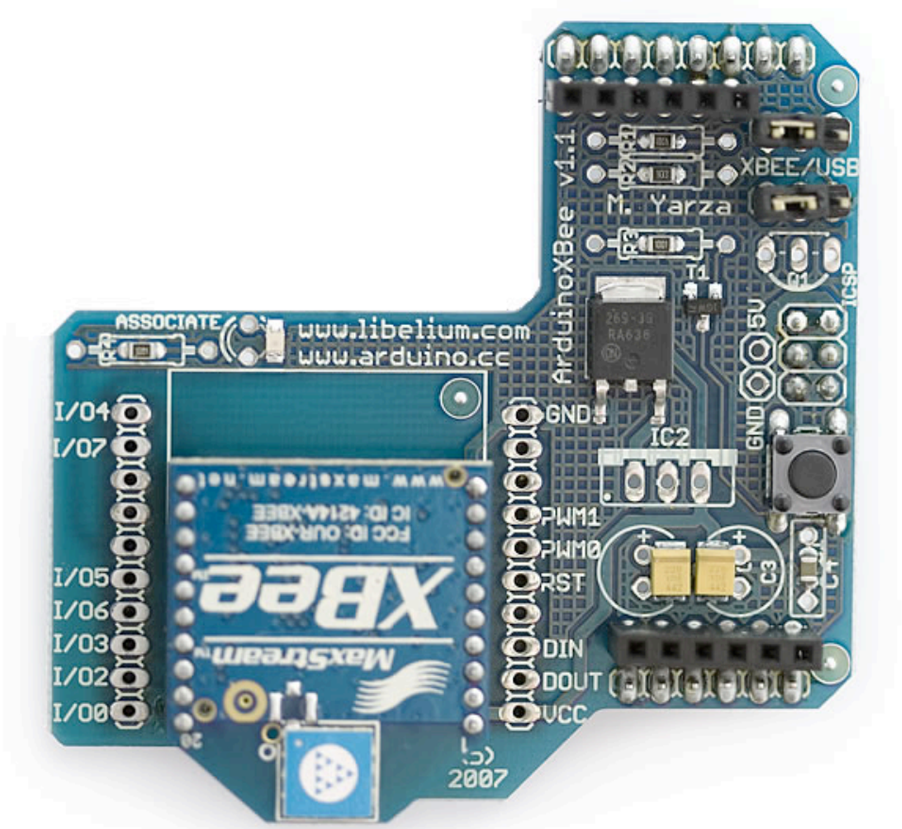
XBee

- I moduli XBee sono ricetrasmettitori digitali in banda 2.4GHz conformi allo standard ZigBee
- Permettono da comunicazione wireless tra centinaia di dispositivi wireless in un raggio di 30m-1Km (la distanza dipende dal modulo utilizzato e dagli ostacoli tra due moduli)
- I moduli possono operare su diverse reti e canali (network e channel ID): è possibile la comunicazione punto-punto e broadcast

LAB Open MediaCenter
SARDEGNA DISTRICT

XBee Shield

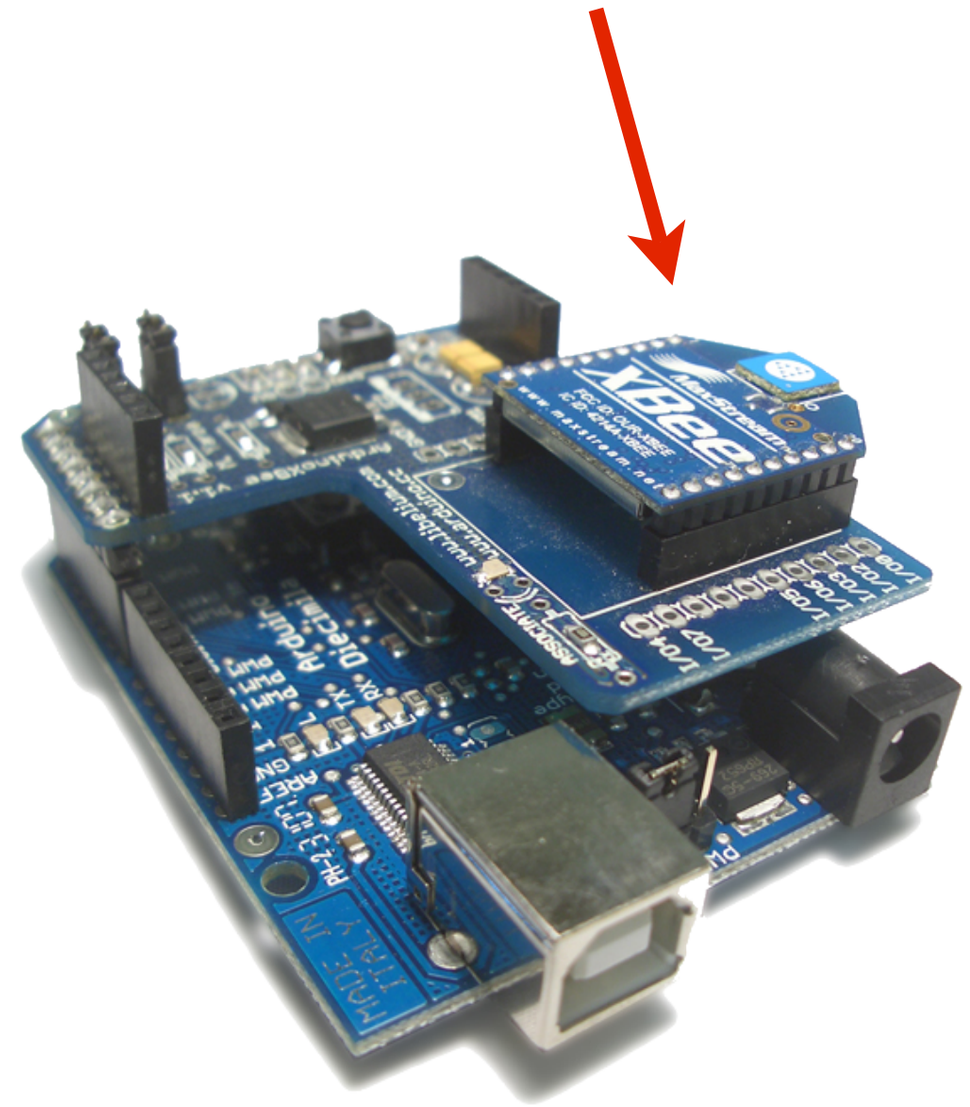
- È una scheda che consente di installare un modulo XBee su Arduino Diecimila
- Interfaccia la porta seriale (PIN 0 e 1) al modulo XBee e provvede alla sua alimentazione



Arduino XBee: note

- Ad esclusione della porta seriale, tutti i PIN sono disponibili per collegamenti a sensori e attuatori
- Per programmare Arduino è necessario rimuovere il modulo XBee

Rimuovere durante la programmazione di Arduino



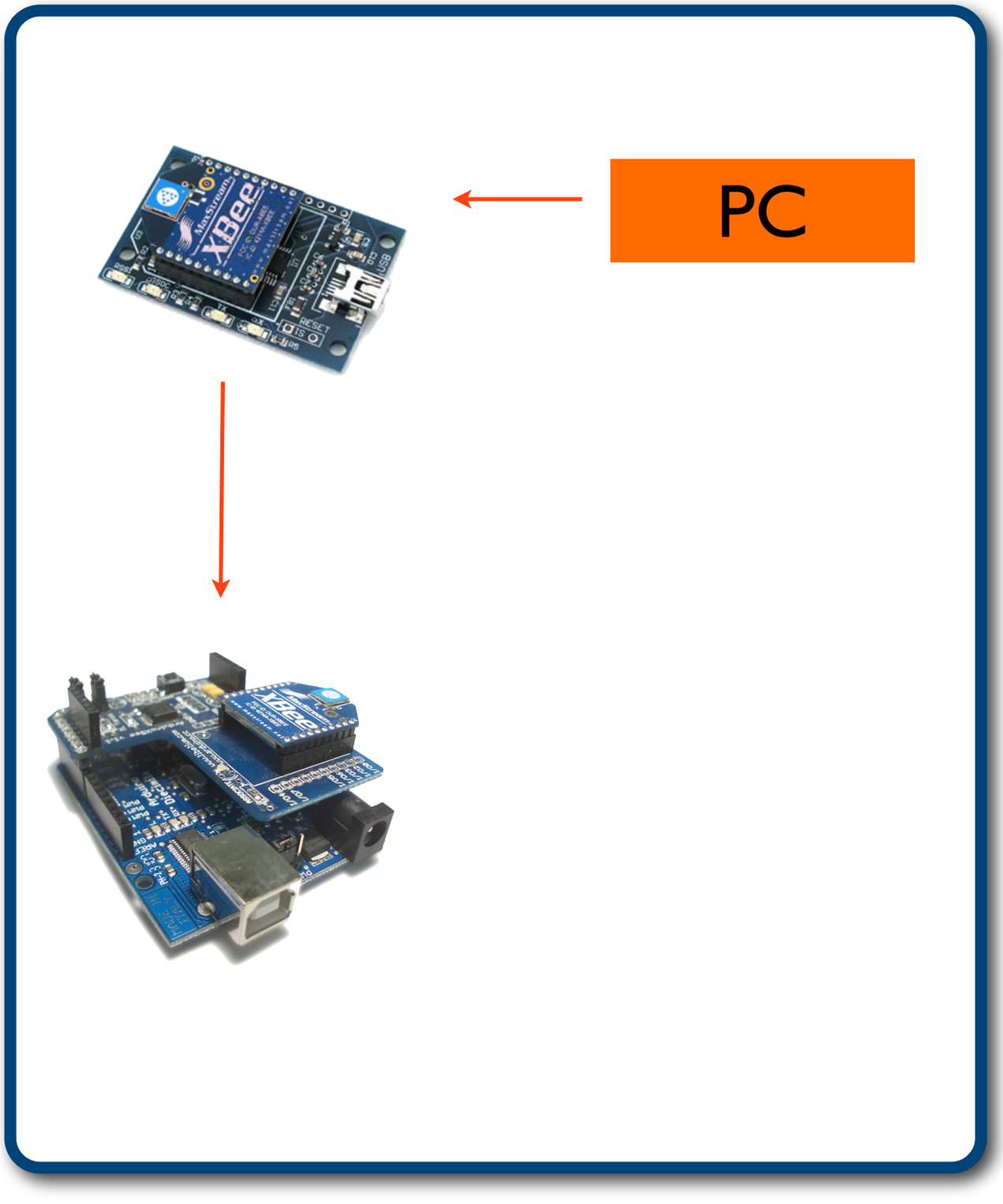
LAB Open MediaCenter
SARDEGNA DISTRICT

XBee USB Adapter

- È un adattatore che consente di connettere un modulo XBee ad un computer attraverso porta USB
- Il modulo XBee è accessibile attraverso la porta seriale (virtuale) associata al modulo

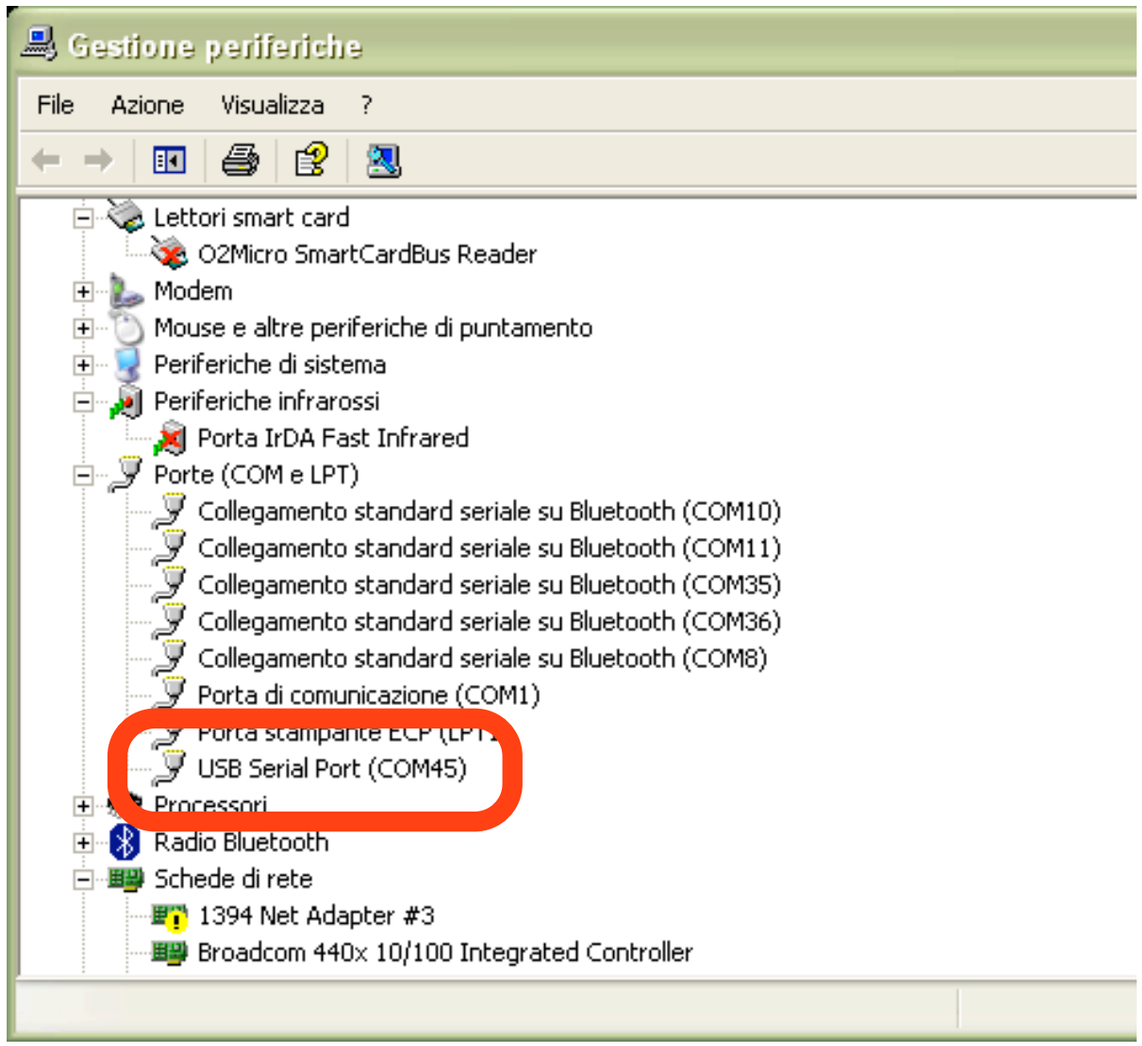
XBee Bridge

- Utilizzando XBee è possibile realizzare una connessione seriale wireless in maniera nel tutto trasparente per l'applicazione



XBee USB Adapter

- La porta seriale assegnata all'XBee USB, è mostrata nel pannello **Gestione Periferiche** del gruppo **Sistema** all'interno del **Pannello di Controllo** di Windows



Hands-on project

Post-it digitale controllato via cellulare Bluetooth

LAB Open MediaCenter

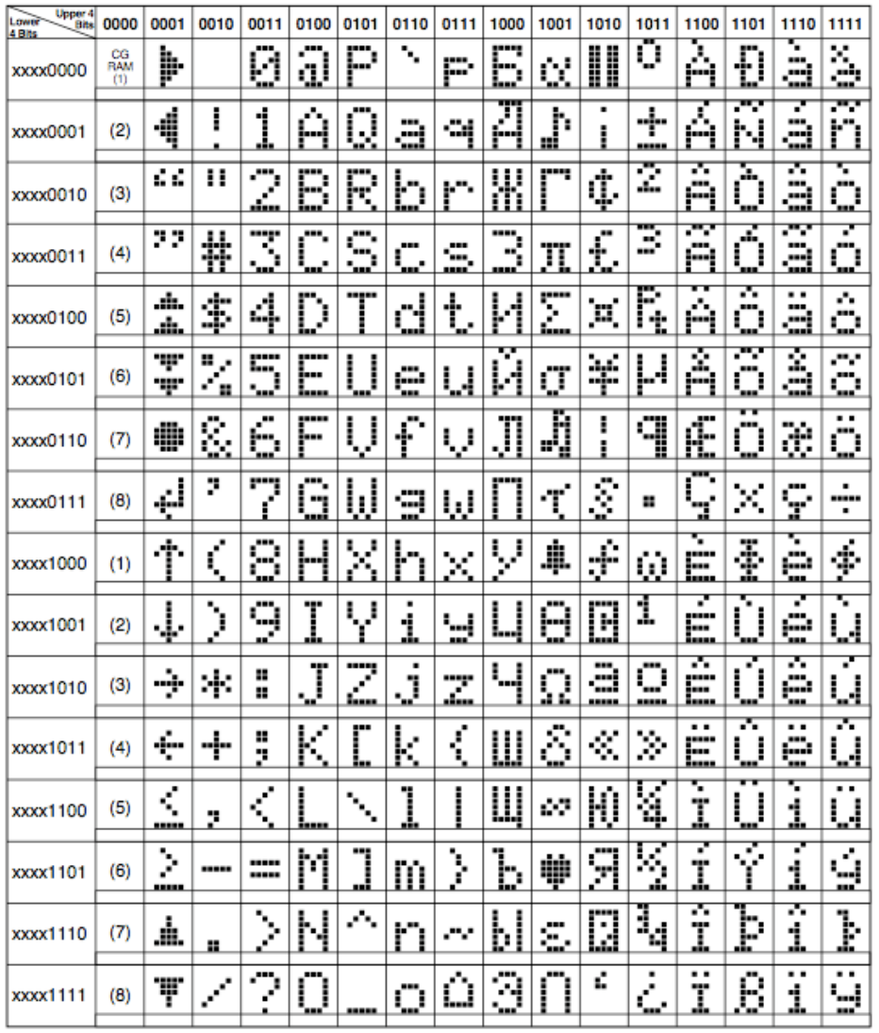
SARDEGNA DISTRICT

CRS4

LAB Open Media Center
SARDEGNA DISTRICT

Display LCD

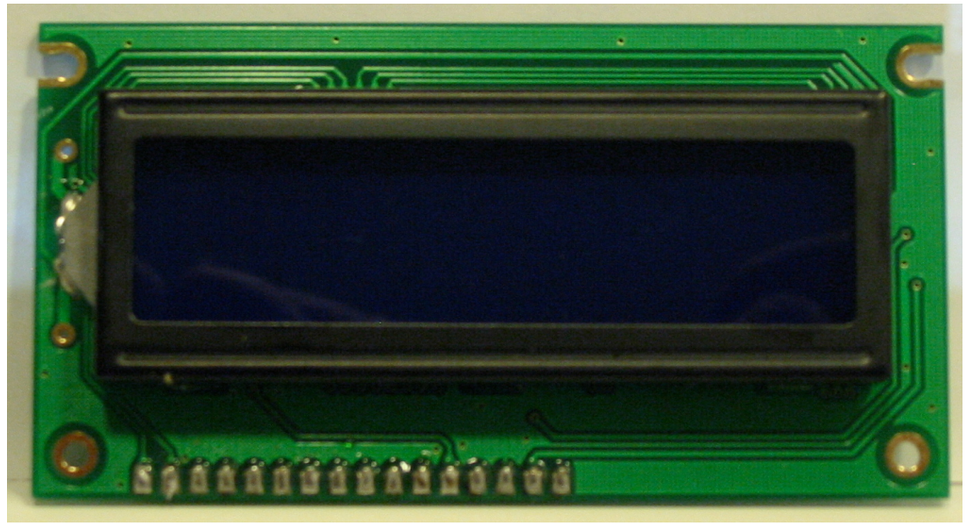
- I display LCD consentono di visualizzare testo su una o più linee, oppure grafica
- Un'apposita CPU intepreta i comandi provenienti dagli input digitali e a pilotare il display

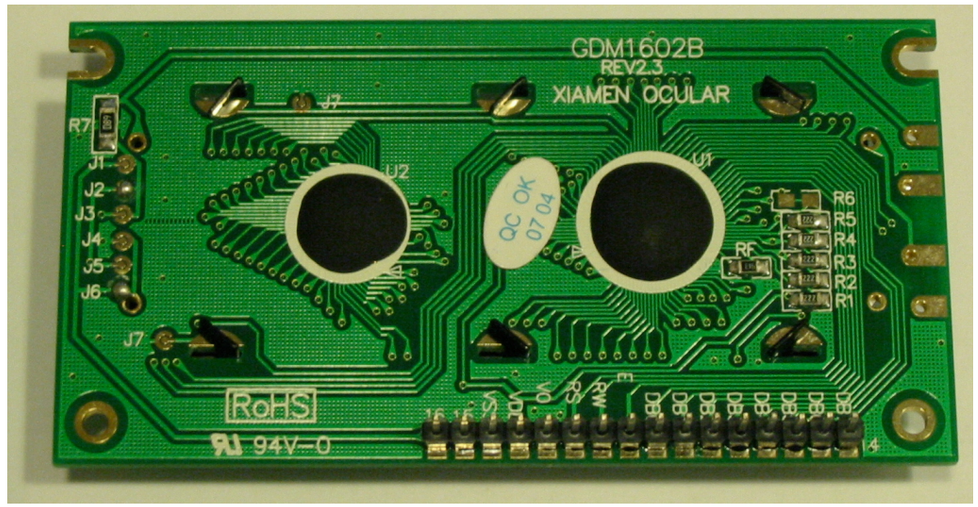


Upper 4 Bits / Lower 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	@	P	`	p								
xxxx0001	(2)		!	1	A	Q	a	q								
xxxx0010	(3)		"	2	B	R	b	r								
xxxx0011	(4)		#	3	C	S	c	s								
xxxx0100	(5)		$	4	D	T	d	t								
xxxx0101	(6)		%	5	E	U	e	u								
xxxx0110	(7)		&	6	F	V	f	v								
xxxx0111	(8)		'	7	G	W	g	w								
xxxx1000	(1)		(	8	H	X	h	x								
xxxx1001	(2)		)	9	I	Y	i	y								
xxxx1010	(3)		*	:	J	Z	j	z								
xxxx1011	(4)		+	;	K	[	k	{								
xxxx1100	(5)		,	<	L	\	l	|								
xxxx1101	(6)		-	=	M	]	m	}								
xxxx1110	(7)		.	>	N	^	n	~								
xxxx1111	(8)		/	?	O	_	o									

Display LCD

- I display più diffusi sono basati sul chip Hitachi HD44780, in grado di controllare modelli di diverse dimensioni (1-4 righe, 8-40 colonne)
- Grazie alla loro diffusione, sono largamente supportati

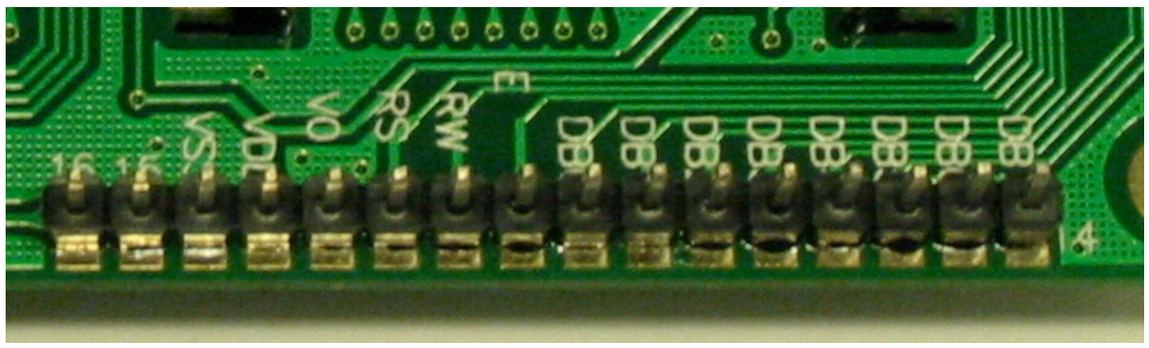




Display LCD

- La disposizione dei PIN può cambiare a seconda del modello: riferirsi al daasheet!
- Occorono:
  - alimentazione 5V+
  - 11 PIN digitali
  - potenziometro (contrasto)

DISPLAY	ARDUINO
DB0 - DB7	3 - 10
E	2
RW	11
RS	12
V0	potenziometro
VDD	5V+
VSS	GND
15-16	LED illum.



LAB Open Media Center
SARDEGNA DISTRICT

LCD Library

- Il modo più semplice per controllare i display LCD è utilizzare la libreria **LCD Library**, che contiene le funzioni per inizializzare il display e inviare stringhe di testo
- La versione disponibile su http://www.arduino.cc non è compatibile con la versione corrente di Arduino e non supporta display con 2 linee
- All'indirizzo **http://www.gerdavax.it/data/LCDLibrary_2x40.zip** è disponibile una versione modificata dell libreria

Installazione LCD Library

- Per poter utilizzare la LCD Library è necessario installarla all'interno dell'ambiente di sviluppo di Arduino
- Occorre decomprimere il file **LCDLibrary_2x40.zip**, che contiene la cartella LiquidCrystal. Occorre copiare questa cartella all'interno della directory **hardware/libraries** contenuta nella installazione di Arduino
- Tale procedura è compiuta una sola volta e non è necessario ripeterla per ogni programma

Display LCD: codice

- Per poter utilizzare la LCDLibrary è necessario "importarla" (grazie alla direttiva #include) all'interno del programma
- Il tipo LiquidCrystal fornisce le funzioni per inizializzare e ripulire il display

```
#include <LiquidCrystal.h>

LiquidCrystal lcd =
LiquidCrystal();

void setup(void)
{
    lcd.init();
    lcd.clear();
}
```

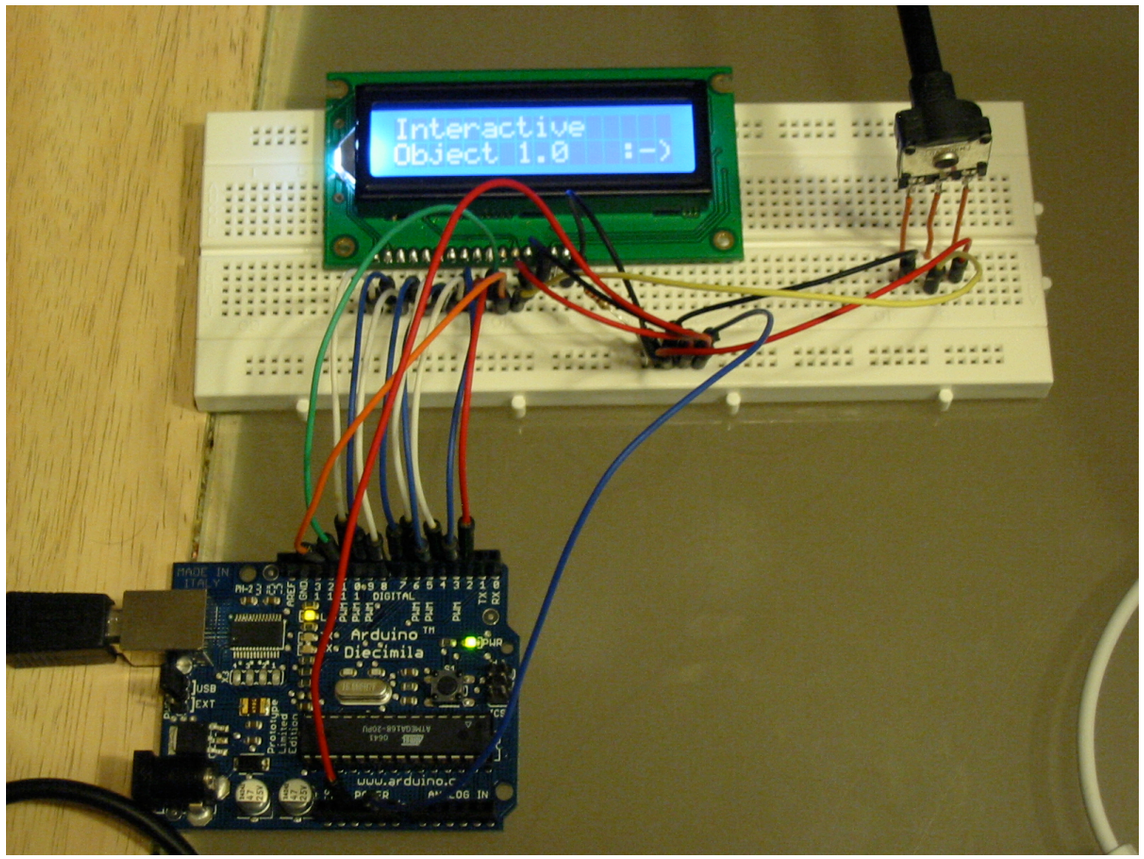
LAB Open Media Center
SARDEGNA DISTRICT

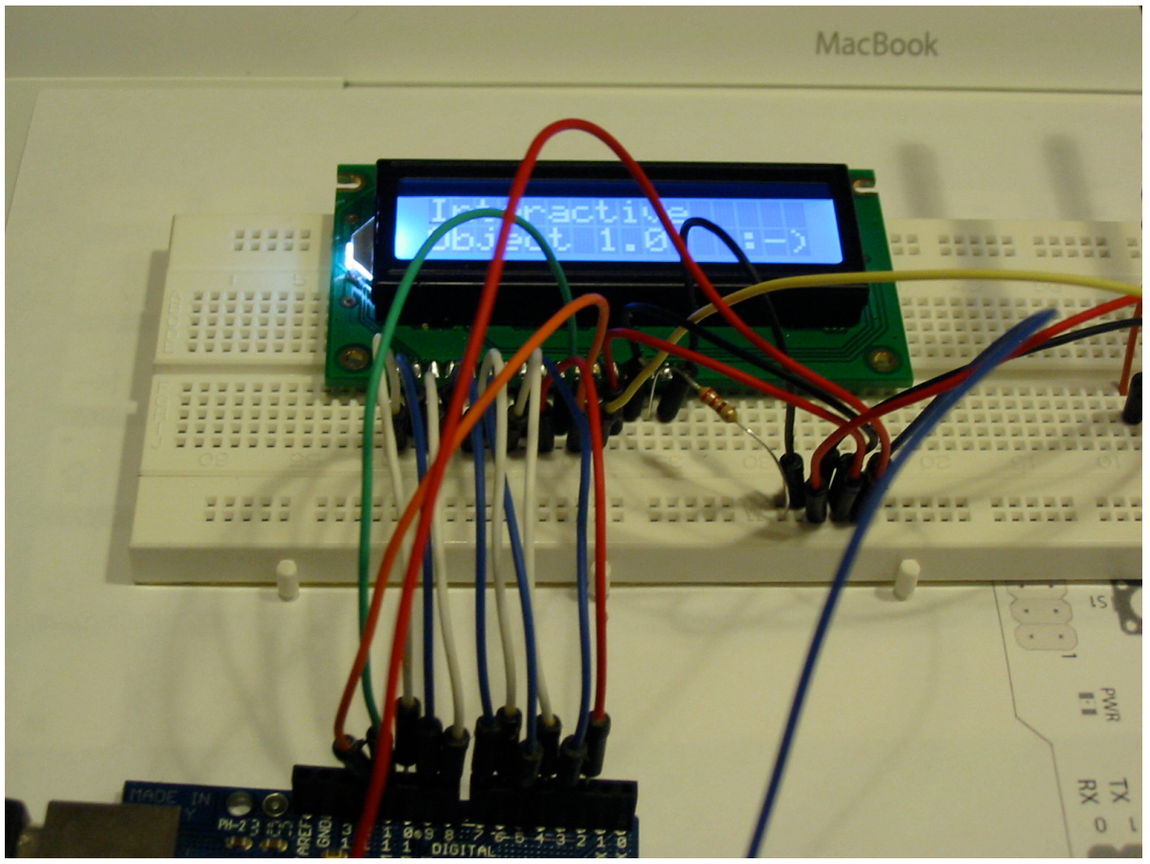
Display LCD: codice

```
void loop(void)
{
    // va all'inizio della prima riga
    lcd.commandWrite(128);

    // scrive nella prima riga
    lcd.printIn("Arduino rocks!");

    // va all'inizio della seconda riga
    lcd.commandWrite(168);

    // scrive nella seconda riga
    lcd.printIn("LCD 2x14 display!");

    delay(1000);
}
```

LAB Open MediaCenter
SARDEGNA DISTRICT

Display LCD

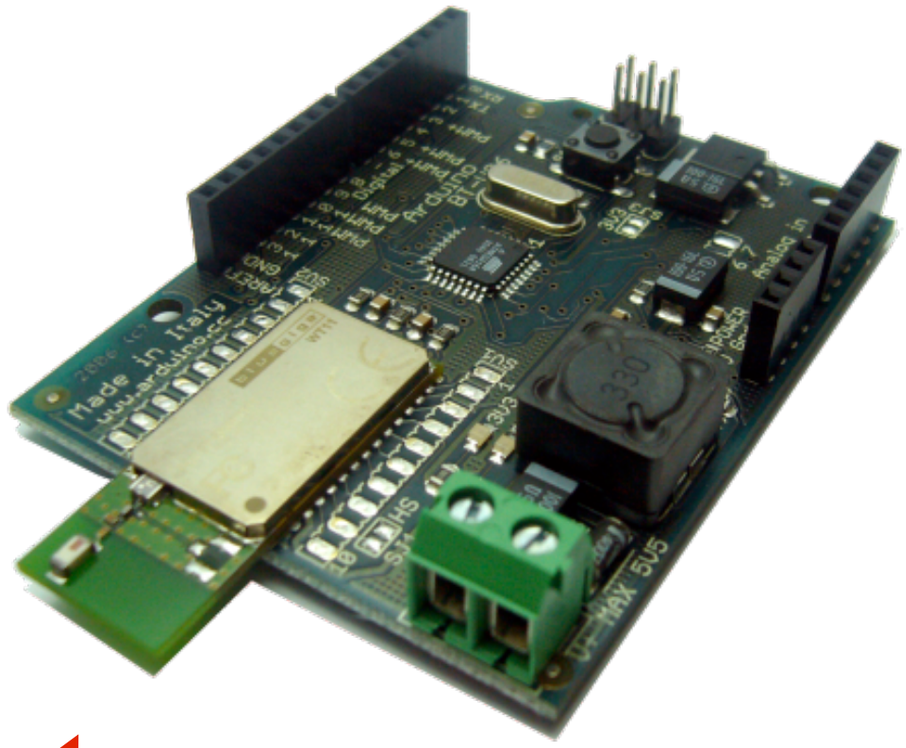




LAB Open Media Center
SARDEGNA DISTRICT

Arduino Bluetooth

- È equipaggiato con un modulo Bluetooth e privo di porta USB
- Programmazione e comunicazione seriale avvengono esclusivamente attraverso connessione wireless Bluetooth

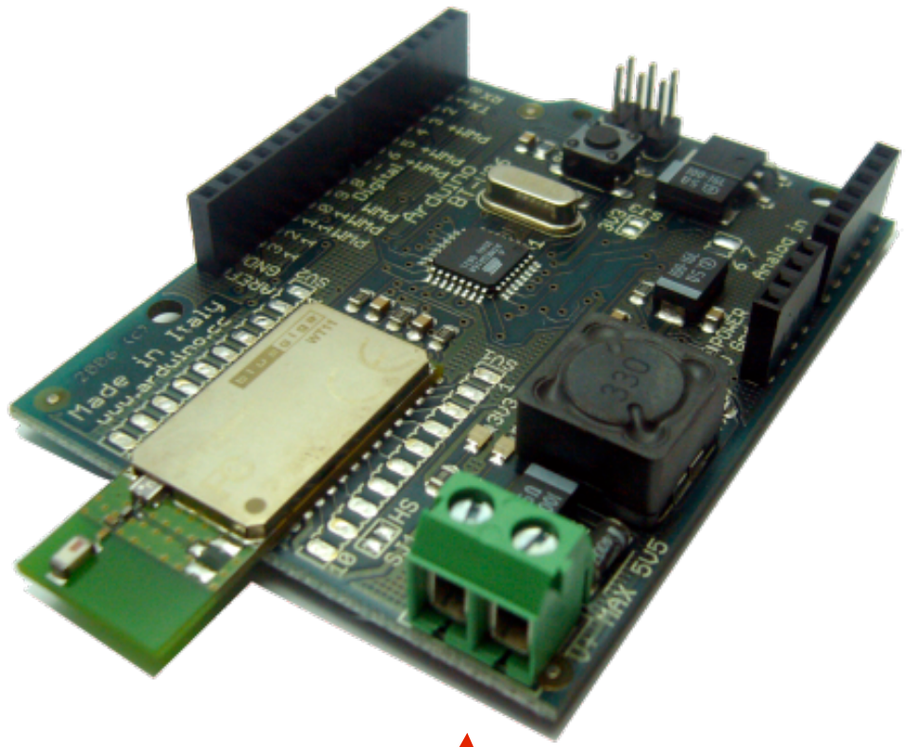


Modulo Bluetooth

LAB Open Media Center
SARDEGNA DISTRICT

Alimentazione

- Non essendo presente la porta USB, è necessario alimentarlo attraverso batterie o un alimentatore
- La tensione massima di alimentazione è 5V attraverso i morsetti



Punto di alimentazione

LAB Open Media Center
SARDEGNA DISTRICT

Java ME e Bluetooth

- Utilizzando un terminale Java ME compatibile con la Bluetooth API (JSR-82) è possibile inviare comandi ad Arduino Bluetooth e realizzare un post-it elettronico con un display LCD
- Il modulo Bluetooth è configurato come slave ed espone una porta RFCOMM
- La velocità della seriale di Arduino deve essere impostata a 115200bps, non essendo possibile impostarla sul terminale mobile

LAB Open Media Center
SARDEGNA DISTRICT

Display LCD: codice

```
void loop(void)
{
    if (Serial.available() != 0)
    {

        lcd.clear();
        delay(10);
        // l = Serial.read() - 48;


        // Serial.println(l);

        for (i = 0; i < 14; i++)
        {
           c = (char) Serial.read();
           lcd.print(c);
        }
```

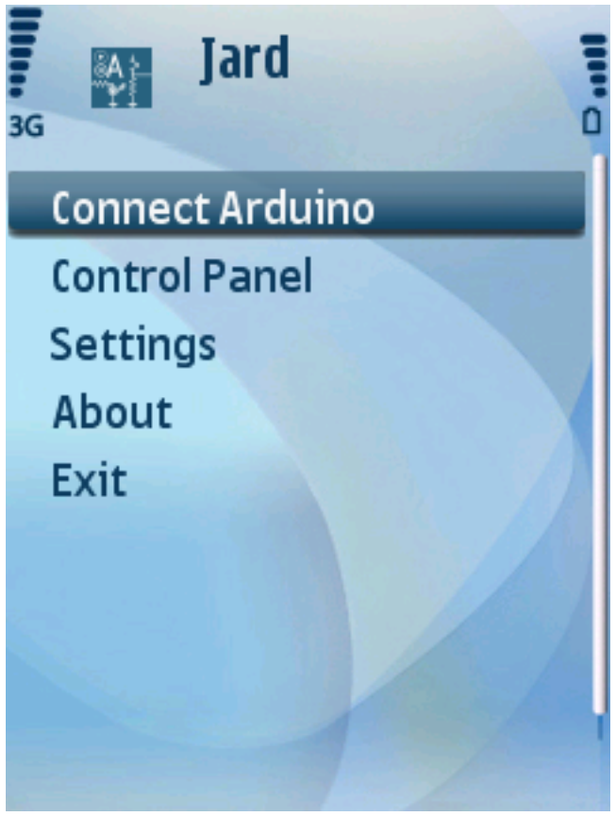
LAB Open Media Center
SARDEGNA DISTRICT

Display LCD: codice

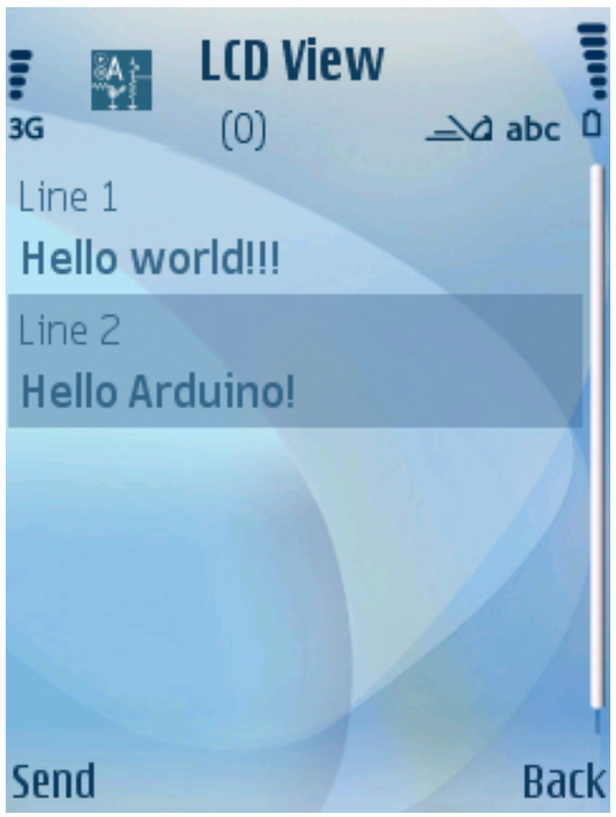
```
lcd.commandWrite(168);

for (i = 0; i < 14; i++)
{
    c = (char) Serial.read();
        lcd.print(c);
    }

}
delay(1000);
}
```

LAB Open MediaCenter

SARDEGNA DISTRICT



JARD



LAB Open Media Center
SARDEGNA DISTRICT

Fonti web

- Arduino Official Site:
  - **http://www.arduino.cc**
- Bionic Arduino:
  - **http://todbot.com/blog/bionicarduino/**
- Limor Ladyada web site:
  - **http://www.ladyada.net/**
- Wikipedia:
  - **http://www.wikipedia.org**
- Dispense del corso: "Laboratorio per Prototipi 1"
  - Stefano Sanna, Università La Sapienza, 2008

LAB Open Media Center
SARDEGNA DISTRICT

Libri

- “Physical Computing: Sensing and Controlling the Physical World with Computers”
  - Dan O'Sullivan, Tom Igoe, Course Technology, 2004
- “Making Things Talk”
  - Tom Igoe, Make Books (O’Reilly), 2007
- “Java Micro Edition - Sviluppare applicationi network-oriented per telefoni cellulari e PDA”
  - Stefano Sanna, Hoepli Informatica 2007
- “Designing for Interaction”
  - Dan Saffer, New Riders, 2007

Arduino hands-on lab

SARDEGNA DISTRICT

LAB Open MediaCenter

CRS4